Investigation of Refractive Bubble Distortions in PIV of Liquid Phase Turbulence in Gas-Liquid Two-phase Flow With Large Bubbles

Undergraduate Honors Thesis
Presented in Partial Fulfillment of the Requirements for
Graduation with Distinction in the
Department of Mechanical Engineering at
The Ohio State University

By: Joshua Jones

May 2012

Advisor: Xiaodong Sun

In Association With: Xinquan Zhou

Benjamin Doup

Abstract

Particle Image Velocimetry (PIV) systems have proven to be a powerful tool for measuring flows under many different conditions. As the technique is based on being able to visualize a flow, there are many inherent issues with its application to complex visualization windows. The refractive effects of light traveling through boundaries between different materials can distort images and produce inaccuracies in PIV measurements. A mathematical analysis of the refractive distortion effects has been done and is outlined in the following document. In this analysis the effects of the setup geometry and bubble inclusions has been analyzed to help with error estimation. First, the effect of the angle and location of light release as well as flow pipe thickness were investigated to see how they affect the image produced. Second the effect of bubbles being situated between the visualization plane and the camera was investigated. Finally, the different components were used to take virtual images of a theoretical bubbly flow and compare them to the base images in order to determine error functions for the bubbles. As an unexpected result, the error was found to be contained within the distorting bubble, providing no unknown error. Because of this, future investigations on reflective effect are suggested to get a better understanding of the distortion of the PIV images.

Acknowledgements

I would first like to thank my advisor on this project, Professor Xiaodong Sun, for all of his guidance on where I should focus my efforts. He was always willing to sit down with me and figure out where I'm at and what I need to do to get results. I know I took a lot of his time at the end trying to tie it all together, but he made time to help me out.

Next I would like to thank Professor Richard Christensen who took time out of his schedule to sit for my defense. Outside of those working with PIV this may not have been a particularly interesting subject so I am grateful that he was willing to be on my committee.

I would also like to thank Xinquan Zhou and Ben Doup. They were willing to let me into their lab in order to work with their setup and discuss my ideas. Xinquan in particular was an invaluable resource for helping me with getting started on the basics of the research. He also was helpful in the formulation of other research leads that do not appear in this document.

Lastly I would like to thank all of my family members and friends who listened to me rant about my research. Talking through the issues I was having throughout the project proved to be invaluable for making progress.

Table of Contents

List of Figures

List of Tables

1. Introduction

Looking at an object in space and identifying its location relative to a viewing location is something that we do every day. It is the basis for how we navigate around our environment and judge the movement of objects relative to ourselves. However, sometimes the simple task of looking at an object and identifying its location can become a challenge when our perception is distorted due to interactions of materials between ourselves and the image. In the method known as Particle Image Velocimetry (PIV) if these distortions are not taken into account the measurements taken by the process can be flawed, making any subsequent analysis and model fitting of that data more challenging.

In flow analysis, the process of PIV uses the concept of looking at an object at a known time and identifying its location. By looking at the same image a short time later it is then possible to calculate the average velocity of the particle. The process is advantaged over some other methods as it is nonintrusive and does not affect the flow that it is trying to capture. Also, the process allows for large viewing windows and full field analysis. Despite these advantages, distortion effects can make the process ineffective unless it is calibrated correctly. Since in flow analysis we need to know exactly how the image we are looking at is changing, if the images aren't calibrated then the changes may be read incorrectly. There are certain methods by which one can resolve the distorted images to get the correct ones (1003012_ImagingTools_D80.pdf, 2011), but under certain dynamic conditions it can be impossible to achieve correctly resolved images.

In the past PIV has been applied to measure flows for low void fractions (Gui, Lindken, & Merzkirch, 1997) with success as compared to other measurement techniques such as Hot Wire and Pitot Tube measurements. There is a gap in its application, however, around the medium void fractions, corresponding to the slug and churn turbulent flow regimes, due to distortion effects. The application of PIV to bubbly flows has helped improve the process in novel ways such as stereo PIV (Arroyo &

Greated, 1991), laser sheet modification (Dinkelacker, Schafer, Ketterle, & Wolfrum, 1992) and color gradation of light (Cenedese & Paglialunga, 1989); however, the advantages of these methods still have not made their way into two phase flow analysis. While methods have been undertaken to minimize the effects of these distortions (Hopkins, Kelly, Wexler, & Prasad), in the preliminary research for this project no correction was found for the distortion from bubbles and application in these medium void fraction flow regimes.

This project is in conjunction with a larger project aimed at capturing a database of two phase flow measurements. As a part of this larger project, this project is specifically aimed at identifying the distortion effects of the bubbles and the potential errors that they create in the images collected. Together with a third project which is using a four-sensor conductivity probe to measure the gas phase in these flows, the three projects are looking at capturing the full velocity fields of vertical two phase flow in a circular pipe with void fractions in the slug and churn turbulent flow regimes. Eventually the goal is to use the database to create models of the flows which can be used in industry and further research.

1.1 Focus of Thesis

This research is specifically aimed at determining the distortion effects of the bubbles in the flow and how they are affecting the visualization of the flow plane. The intent is to provide a method for quantifying the potential error due to bubbles being present in the flow. By comparing the dispersion profiles of light traveling through a computer modeled setup with various dimensions and bubble conditions this project will provide insight into these potential errors. By knowing more about the effects of varying parameters, visualization setups can be modified to help improve the accuracy of the measurements taken of it.

1.2 Significance of Research

Due to the random sizes, shapes and locations of the bubbles in turbulent flows with high void fractions, there is no way to fully calibrate a PIV system prior to collecting data. Unknown distortions of the images due to these bubble interactions can make particles appear to be not where they actually are. This location shift causes errors in traditional processing of the image. In order to be accurate in reporting results one must have an estimate of the error they could have. Knowing this error makes it easier to understand what may be happening and verify results or match models to the flow.

1.3 Overview of Thesis

For the purpose of identifying errors in the measurements a program was created to mathematically map the light from its emitted location and direction onto the camera plane. Chapter 2 starts this analysis off with the description of the particular setup being used and the simplifying assumptions of how the flow was recreated. In chapter 3 the formulation of the program for the geometrical mapping of light through the system is described, including the modifications of the analysis for bubble inclusions of varying locations and sizes. Chapter 4 then describes how this mapping was used to gain an understanding of the resulting errors and distortion profiles for the various conditions based on how the distorted images reached the camera. Chapter 5 takes a qualitative look at the results and discusses their significance. Chapter 6 then summarizes the findings and presents suggestions for future investigation.

2. Setup Description and Simplifying Assumptions

This research is focused on measuring flow in a vertical pipe in the slug and churn turbulent flow regimes. In order to create this flow in the lab a test rig was constructed from acrylic which uses

pressurized air to create concurrent bubbles as water is piped up into the bottom of the pipe.

Schematically this setup is shown below in figure 1.

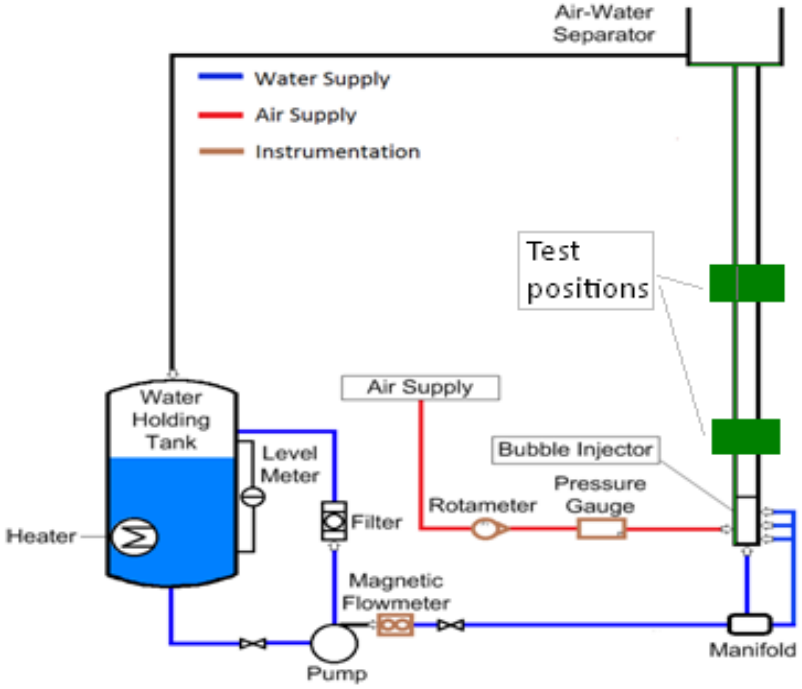

Figure 1: Schematic of experimental setup being used to simulate flows (Zhou & Sun, 2011)

The key dimensions of this setup being used are as follows, although the can be easily modified in the code to analyze different test geometries.

Table 1: Specific setup parameters

Item	Material	Specification
Main Flow Pipe	Acrylic	49 mm inner dia. 57.5 mm outer dia.
Viewing Box	Acrylic	152mmX152mm outer side lengths 17 mm wall thickness filled with water
Structures	Acrylic	index of refraction = 1.55
Liquid Component	Water	index of refraction = 1.33
Gas Component	Air	index of refraction = 1.00
Camera	NA	4 mega pixel 14 bit CCD LaVision at 600mm from viewing box 62mm aperture

For simplicity it was assumed that the pipe was perfectly round, properties of materials were uniform and constant as stated, and distances and measurements were exactly as specified. Further, it

was assumed that the bubbles were spherical and do not intersect each other or the pipe wall. Along with this the concept of symmetry was applied when applicable to make the calculations quicker and more reliable due to potential sign issues in the trigonometry. Also, particles emitting light were assumed to be only in the fluid component of the viewing plane. Lastly it was assumed that any light that was reflected was entirely dissipated and did not reach the camera plane.

3. Program Formulation

For tracking the paths of light traveling through the various flow conditions a section of the pipe perpendicular to the plane was investigated. Bubble effects were analyzed with respect to their distortion in this cross sectional plane. For the vertical effects the code can be set to have an extremely large pipe radius, while maintaining the range of the analysis to just the center of the flow within the actual pipe size. This would imitate the perpendicular nature of vertical intersection that would be seen by the light, but would not be entirely perfect as the amount of material that the light would pass through would vary with location and the distances between walls would be different.

The tracking of the flow was done using mainly the following four equations with other basic geometrical and trigonometric relationships and rules.

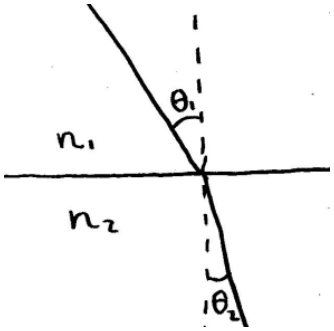

Figure 3: Snell's Law

Equation 1: Snell's Law

$$\eta_1 * \sin\theta_1 = \eta_2 * \sin\theta_2$$

Equation 2: Law of Sine's

$$\frac{\sin A}{a} = \frac{\sin B}{b} = \frac{\sin C}{c}$$

Equation 3: Law of Cosines

$$c^2 = a^2 + b^2 - 2 * a * b * \cos(C)$$

Equation 4: Sagitta Cord Length Formula

$$S = r - \sqrt{(r^2 - l^2)}$$

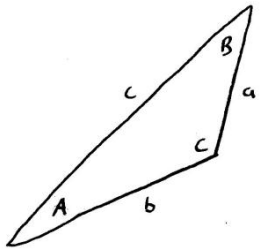

Figure 2: Law of Sines and Cosines

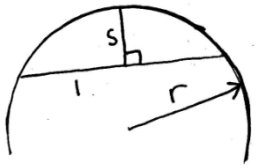

Figure 4: Sagitta Chord Length Formula

By carefully applying these equations, the light path geometry through the system was calculated for the various flow conditions. If at a boundary the light ray were calculated to internally reflect then the release point was marked with a 90 deg. angle for all angles and 90 in. for all distances.

3.1 System Geometry Analysis

Initial tracking of the image distortion started with simply recreating the distortion from the known dimensions of the experimental apparatus with no bubbles present. For this analysis a cross section of the flow was used as shown in Figure 5 on the next page. By using the key equations, along with basic trigonometric and geometric relationships the key variables were found.

The following is a summary of the resulting relationships.

$$\alpha = \sin^{-1}\left(\frac{P}{R} * \sin\theta\right) \quad (1)$$

$$B = \sin^{-1}\left(\frac{\eta_1}{\eta_2} * \sin\alpha\right) \quad (2)$$

$$\gamma = \sin^{-1}\left(\frac{R}{R + t_p} * \sin(180 - \beta)\right) \quad (3)$$

$$\delta = \sin^{-1}\left(\frac{\eta_2}{\eta_3} * \sin\gamma\right) \quad (4)$$

$$\varphi = 180 - \theta - \alpha \quad (5)$$

$$d_{in} = R * \cos(\varphi) \quad (6)$$

$$\varsigma = \beta - \gamma \quad (7)$$

$$d_{pipe} = (R + t_p) * \cos(\varphi + \varsigma) - d_{in} \quad (8)$$

$$d_{po} = (R + t_p) * \sin(\varphi + \varsigma) \quad (9)$$

$$\varepsilon = 90 - \varphi - \varsigma - \delta \quad (10)$$

$$\eta = \sin^{-1}\left(\frac{\eta_3}{\eta_4} * \sin\varepsilon\right) \quad (11)$$

$$\xi = \sin^{-1}\left(\frac{\eta_4}{\eta_5} * \sin\eta\right) \tag{12}$$

$$d_{wall} = t_w * \tan(\eta) \tag{13}$$

$$d_{final} = t_w * \tan(\xi) \tag{14}$$

$$d_{box} = \left(d_w - d_{po}\right) * \tan(\varepsilon) \tag{15}$$

$$d_{total} = d_{in} + d_{pipe} + d_{box} + d_{wall} + d_{final} \tag{16}$$

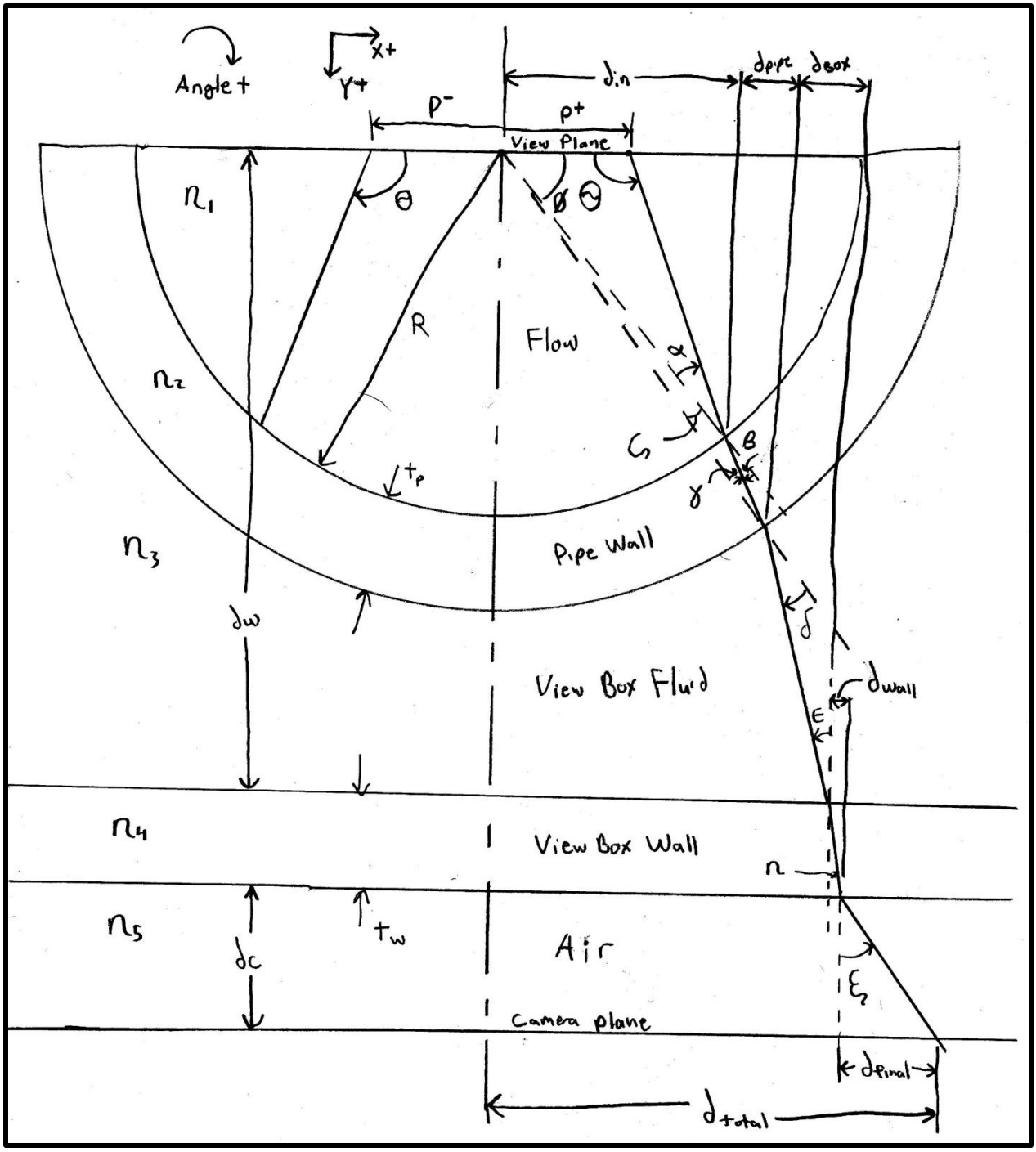

Figure 5: Schematic of setup geometry with sign conventions and definitions of key angles used. Note on the left half the variable θ is defined opposite to how it is defined on the right. This illustrates how the mirroring was done for the analysis.

3.2 Bubble Distortion Analysis

Following the analysis for tracking the light through the main pipe and viewing window the program was modified to see how the addition of different bubbles affected the output of the light. To do this a function was added which took the input light and decided if that light would pass through the bubble that was added to the flow. By knowing the size of the bubble, its location, the release point and angle of the light and the materials of the bubble and surrounding fluid, this program applied the fundamental equations to trace the light through the bubble. By calculating the final location within the pipe at the exit of the bubble and the final angles with respect to the viewing plane a new release point and angle were given to the program to mimic what effect the bubble had. This method would be able to handle multiple bubbles as long as care is taken to account for light distorted by one bubble which then intersects a second bubble. The main equations for this are shown below with Figure 6 showing the angle definitions used.

$$\mathrm{cd} = \sqrt{\mathrm{dby}^2 + (\mathrm{dbx} - \mathrm{P})^2} \tag{17}$$

$$\mathrm{L} = \tan^{-1}\left(\frac{\mathrm{dby}}{\mathrm{dbx} - \mathrm{P}}\right) \tag{18}$$

$$\mathrm{M} = 180 - \theta - \mathrm{L} \tag{19}$$

$$\mathrm{N} = 180 - \sin^{-1}\left(\frac{\mathrm{cd}}{\mathrm{r_{bub}}} * \sin \mathrm{M}\right) \tag{20}$$

$$\mathrm{H} = 180 - \mathrm{N} \tag{21}$$

$$\mathrm{I} = \sin^{-1}\left(\frac{\eta_{\mathrm{fluid}}}{\eta_{\mathrm{bub}}} * \sin \mathrm{H}\right) \tag{22}$$

$$\mathrm{J} = \mathrm{I} \tag{23}$$

$$\mathrm{K} = \sin^{-1}\left(\frac{\eta_{\mathrm{bub}}}{\eta_{\mathrm{fluid}}} * \sin \mathrm{J}\right) \tag{24}$$

$$\mathrm{O} = 180 - \mathrm{J} - \mathrm{I} \tag{25}$$

$$\theta^* = \theta - 180 + \mathrm{O} + \mathrm{H} + \mathrm{K}) \tag{26}$$

$$s1 = \sqrt{\left(cd^2 + {r_{bub}}^2 - 2cd * r_{bub} * \cos(H - M)\right)} \quad (27)$$

$$s2 = \frac{\sin(O)}{\sin(I)} * r_{bub} \text{ *** This is sensitive to inflection} \quad (28)$$

$$x = \cos(M + L) * s1 + \sin(270 - N - M - L - I) * s2 \quad (29)$$

$$y = \sin(M + L) * s1 + \cos(270 - N - M - L - I) * s2 \quad (30)$$

$$xa = \frac{y}{\tan(\theta^*)} \quad (31)$$

$$P^* = P + x + xa \quad (32)$$

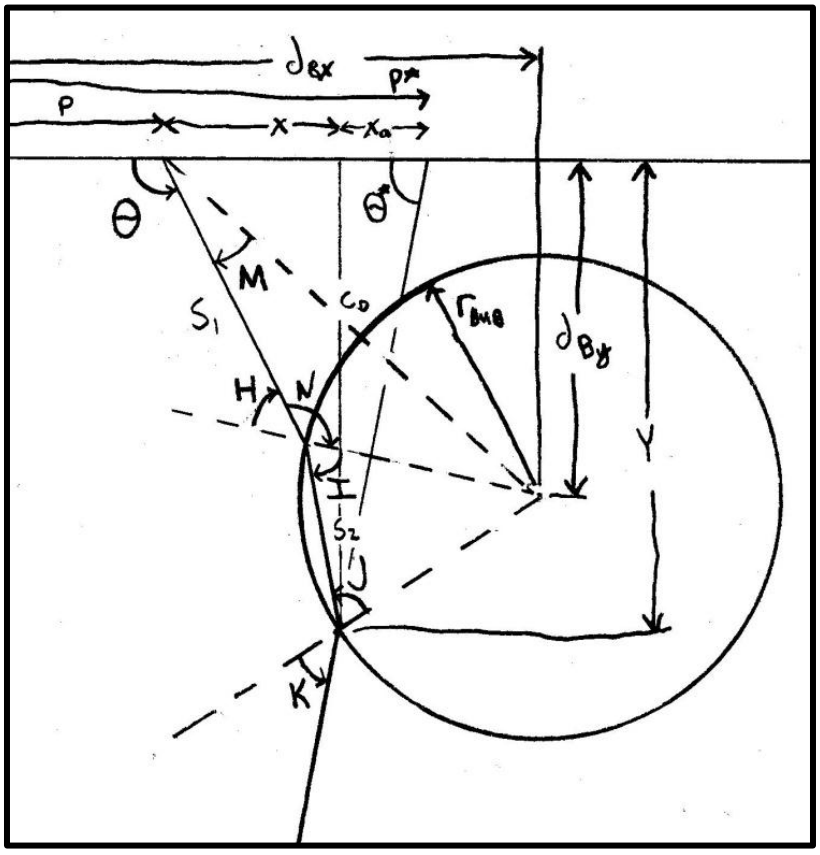

Figure 6: Bubble geometry. Due to the nature of the solution there can be ambiguous cases for some of the calculations. This will manifest in nonsensical results or unexpected errors and should be consitered when using the program.

Since the light doesn't always intersect with the bubbles it was also necessary to determine which location and angle combinations intersected with the bubble. For bubbles intersecting the plane, particularly for slug bubbles, it was necessary to identify where light was actually being emitted from. Since in the experimental setup only the fluid portion of the flow is seeded, only that phase can emit

light, thus, when the release point was found to be inside of a bubble it was marked so that it could be removed from the data set later and didn't affect other calculations. The following equations and figure provide the basis for determining if the light would pass through a bubble and if it was actually released from the fluid phase. These equations were used to determine the "shadow" range of release conditions that would intersect the bubble.

$$d_{left} = dbx - abs\left(\frac{dby}{\tan(180-\theta)}\right) - abs\left(rbub * \cos(\theta - 90)\right) \quad (33)$$

$$d_{right} = dbx - abs\left(\frac{dby}{\tan(180-\theta)}\right) + abs\left(rbub * \cos(\theta - 90)\right) \quad (34)$$

$$l = \sqrt{2 * r_{bub} * s - s^2} \quad (35)$$

$$di_{left} = dbx - l \quad (36)$$

$$di_{right} = dbx + l \quad (37)$$

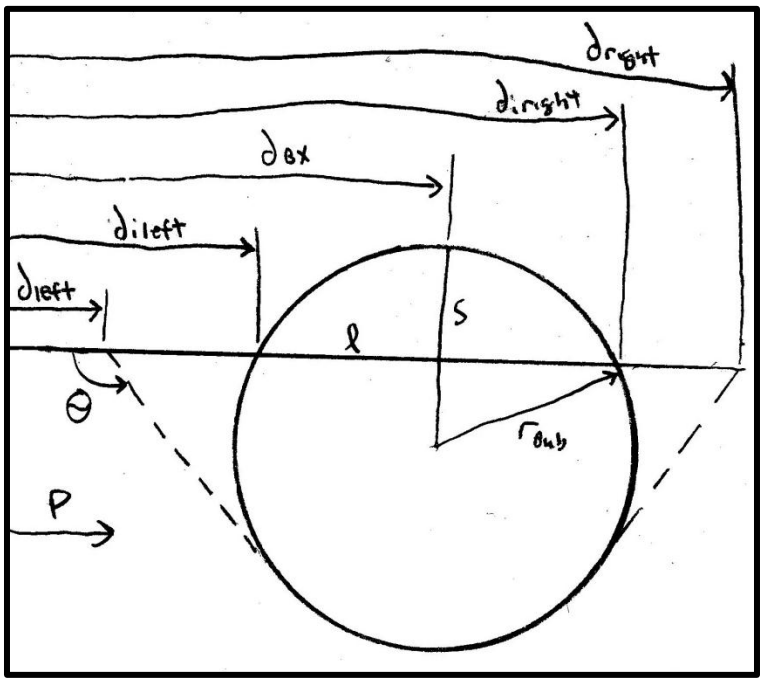

Figure 7: Bubble "shadow" geometry

3.3 Error Calculation

The determination of the error amount caused by bubbles in the flow was done by first taking the base image cast through the pipe with no bubbles that would be picked up by a the camera and finding its calibration profile. This calibration profile was created by taking the initial release distances and subtracting the resulting distorted distances. This profile, when added to the distorted images should then be able to put the image back to its theoretically correct values; just as a calibration of the system would in actual application. By applying the calibration to the images with different bubble inclusions and then subtracting that result from the release profile an offset distance was found. This offset distance data was used to then analyze the potential measurement error.

Figure 8 below shows conceptually how the initial release profile was found in order to create the correct "focused" image profile. This was done by making all rays coincident at the center point of a lens representative of the cumulative lens effects of all structural components. After the release arrays were found, by sending the image a distance of fc past the common point, the image profile was recreated for the image. It was assumed that the converged point was at the center of the camera lens.

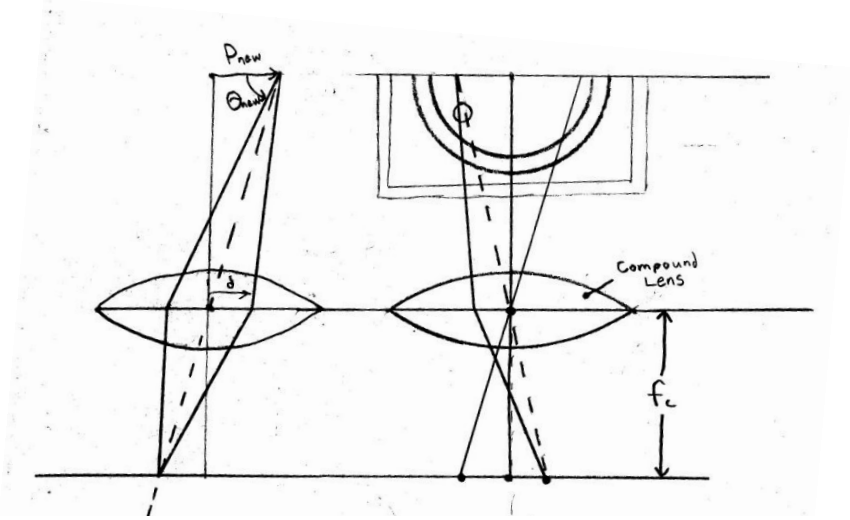

Figure 8: Calibration concept

If rays were determined to not go through the center of the lens after being distorted they were adjusted to account for the refractive effect of the camera lens. Since the system is treated as a single compound lens, the image had to start at the initial viewing plane in order to be able to use the same focal parameters. To do this the distorted release point from the bubble program was used to recalculate the new release parameters which would intersect the common point used initially. If the setup is in focus the resulting distorted location would then be calculable using the additional focal distance as was done for non-distorted points.

4. Results and Discussion

From the mapping of the light through the system and using the equations and sign conventions found in Figure 5 to **Error! Reference source not found.**; the codes that can be found in appendix A were created. Using these codes it was found that, for the basic geometry being used, the sizing of system geometry only really affected the potential for internal reflection of the system with minimal effect on the amount of error induced by the bubbles after calibration was completed. Whenever bubbles were added to the system it was found that there were portions of their profile that would experience internal reflection phenomena and thus cause some of the profile to be lost. Also, if bubbles intersected the viewing plane, portions of the image would be blank as the fluid would not be there to emit light. Between the loss of light due to bubbles intersecting the light plane and the effects of internal reflection in the sides of bubbles that do have light passing through, very little dispersion that could be misinterpreted was found in the single PIV mode. This is because, when the light was of sufficient parameters to travel through the system under the setup conditions, the amount of measurement error was not large enough where the point would actually appear outside of the bubble. In theory any errors that would be present would be easily identifiable by halos with no images in them. Within the halos in theory an estimate of the measurement could then be constructed with the maximum error being the

particle moving from one extreme to the opposite extreme of uncertainty in measurement; however in practice these areas could also simply be removed from the analysis.

4.1 System Geometry Effects

When the program was run without bubbles in the system, maps of the effects of the various parameters were created. In particular it was found that the angle of light release, the distance it was released from and the thickness of the wall portions of the setup affected the distortion of the images the most. The following figures show the final angles and distance offsets as a function of these different variables. Figure 9 and Figure 10 below show the effect of release angle and release distance on both the final angle and displacement. From these two plots we can see that if a point is in the positive release direction then adjusting the release angle to be smaller acts to cause the point to be sent back to the center of the tube. On the other hand, if the release angle is increased then the image of the negative release points is sent back towards the center of the tube. This phenomenon will help in getting the image to land entirely on the camera aperture opening of 62mm and subsequently the CCD chip itself. By varying the release angle as a function of the release distance across the entire viewing plane the gradient that mimic what would be seen by the camera can be calculated. This gradient allows us to calculate the focused images, and then numerically correct the images that are produced and in doing so produce a calibration overlay field that can be applied to distorted readings to show the distortions seen despite the calibration, and thus the overall error.

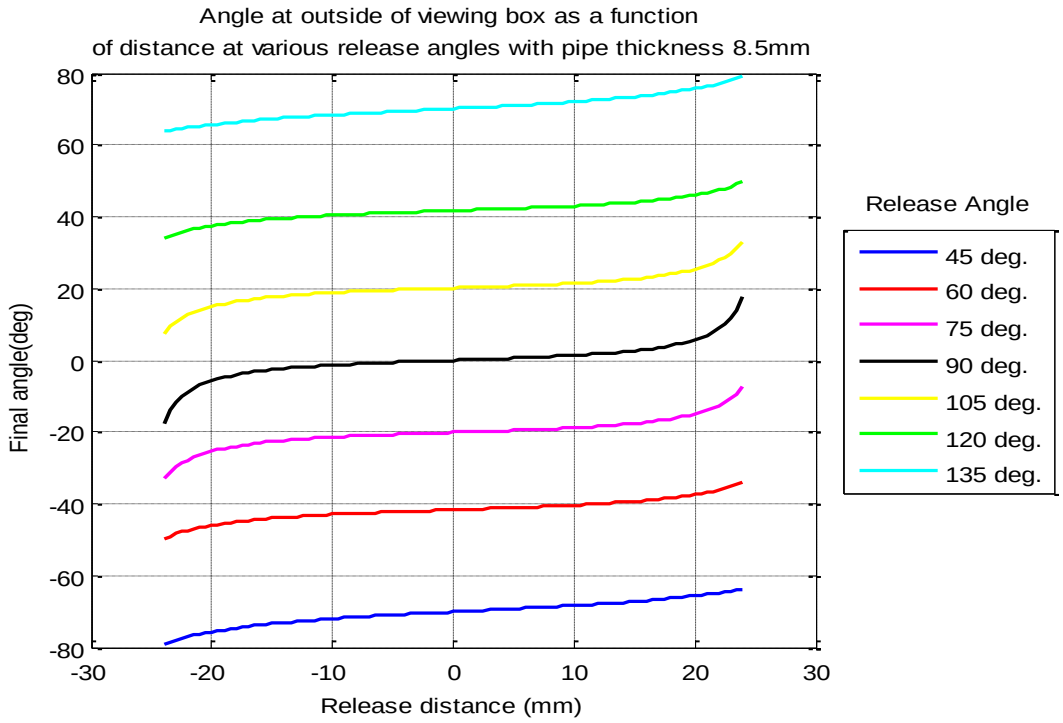

Figure 9: Final angle (ξ) as a function of release distance and angle.

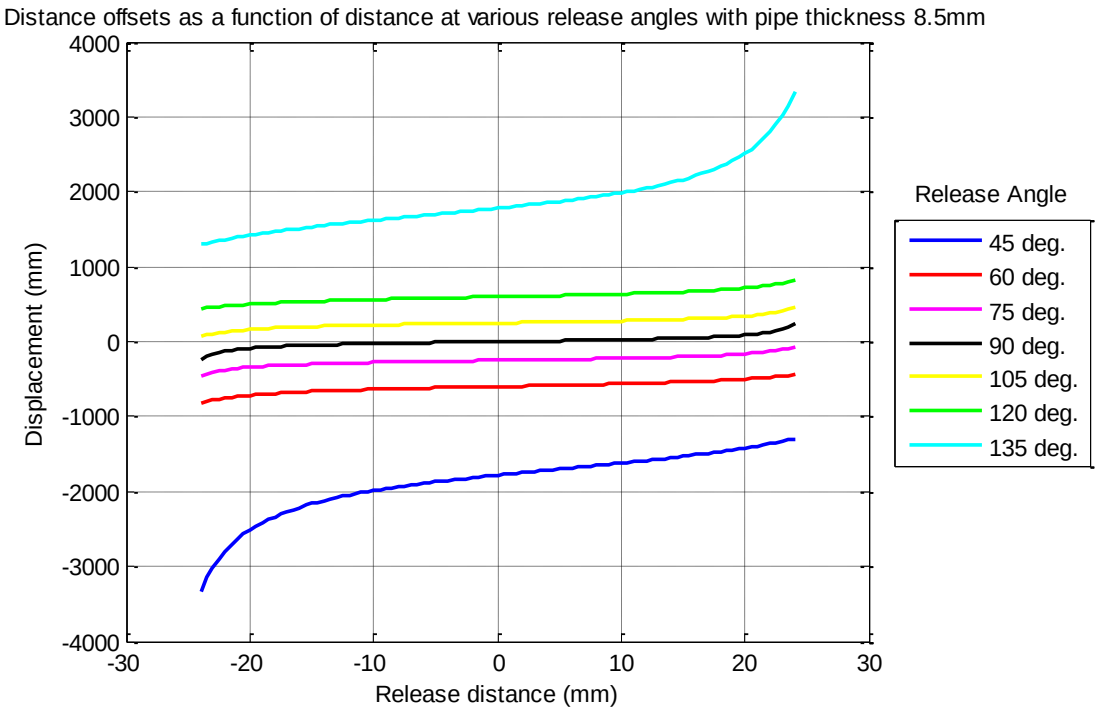

Figure 10: Displacement (dtotal) as a function of release distance and angle.

Next, by looking at the opposite scenario, where the release distance is held constant and the release angle is varied we get another look at how the system behaves. When doing this calculation particular attention had to be given to the acceptable release angles as angles that are too large can cause the light to internally reflect at the boundaries. The results of these calculations are shown in Figure 11 and Figure 12. It can be noted from these plots that as the angle is taken to an extreme the curves become more similar despite release location. This may be useful when analyzing stereo PIV systems which rely on angled viewing for determining components of the flow as the amount of bubble error may be scaled based on the angle of viewing.

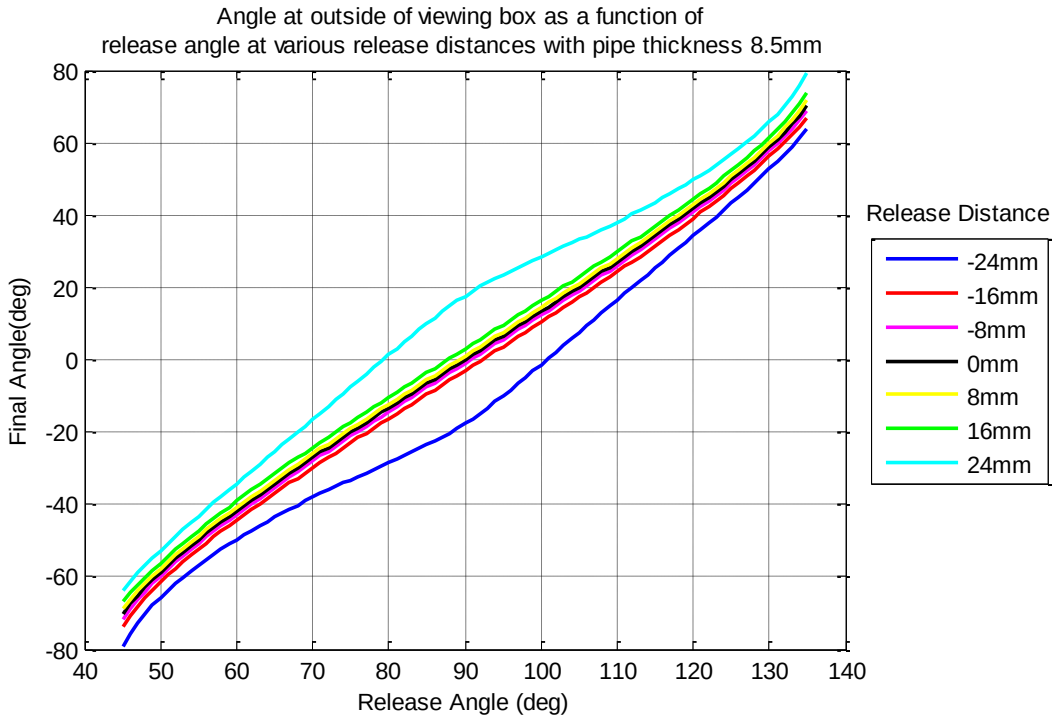

Figure 11: Final angle (ξ) as a function of release angle and distance.

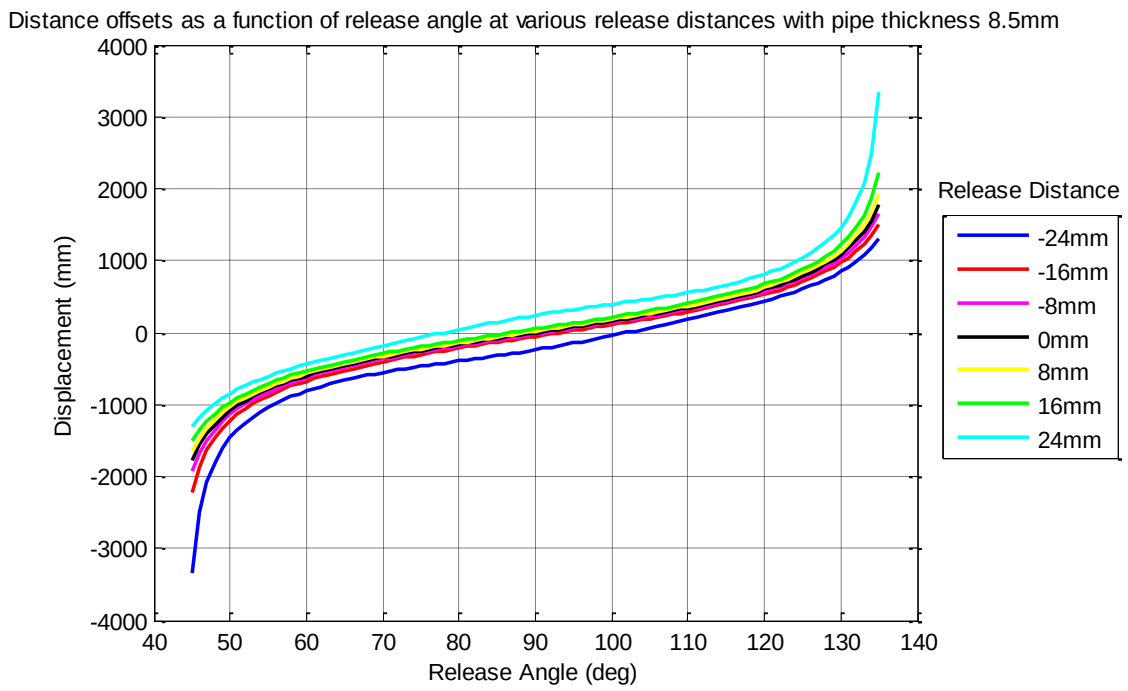

Figure 12: Displacement (dtotal) as a function of release angle and distance.

The final system parameter that was investigated was the effect of increasing the pipe wall thickness. This was expected to cause the displacement error in the measurements to be greater due to a longer travel distance on a distorted path. It can be seen from Figure 13 and Figure 14 that increasing the thickness of the pipe did increase the displacement seen by the system, but will be seen later to not greatly affect the errors seen in measurements due to the initial calibration of the system.

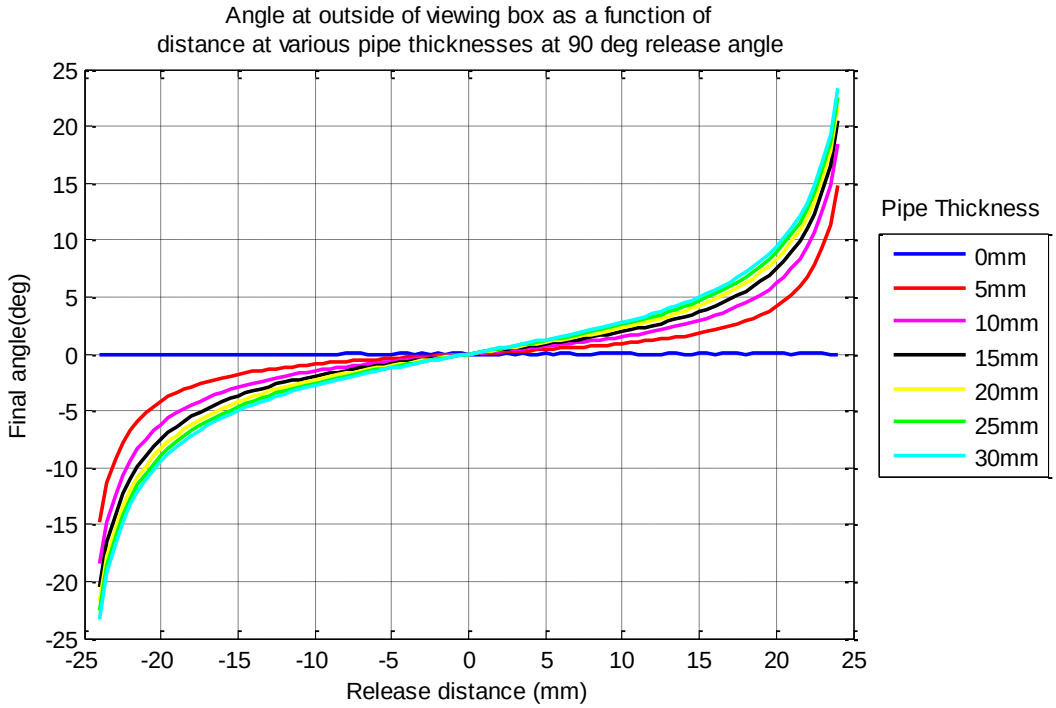

Figure 13: Final angle (ξ) as a function of pipe thickness and distance.

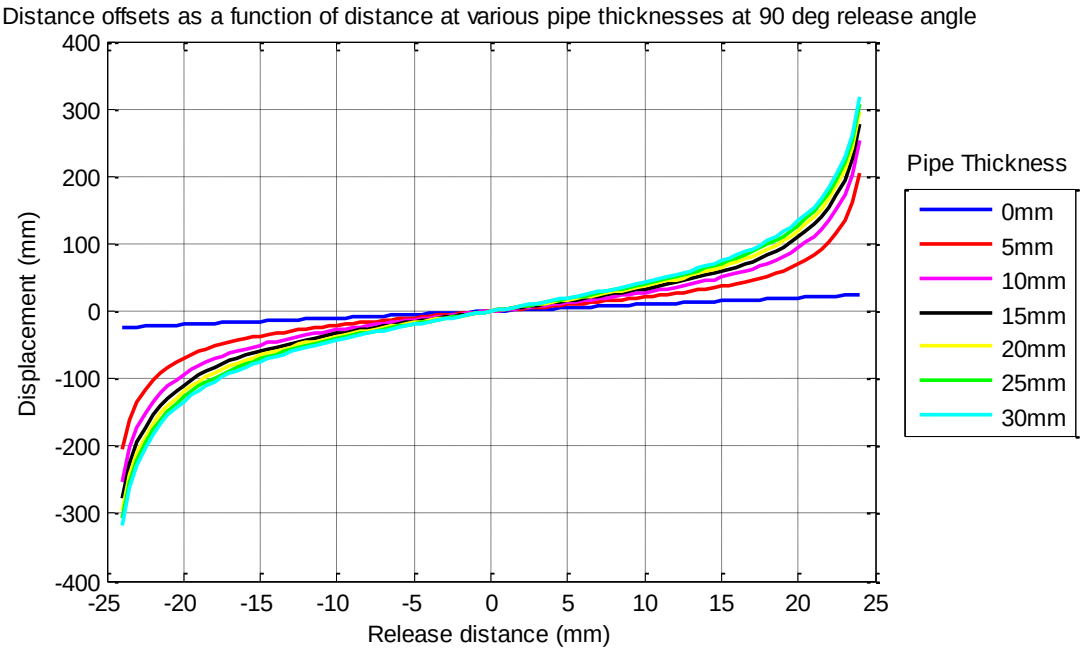

Figure 14: Displacement (dtotal) as a function of pipe thickness and distance.

As mentioned previously, as a last portion of the system analysis, the light was traced back to the camera's CCD chip. This was done by numerically sending all light rays back to a single point in the

center of the camera lens. This allowed the image to be treated as though it were focused by the camera at a focal distance sufficient to get the image the correct size of the systems CCD chip. When a ray was calculated to pass through this center point it was designated as the release angle for the given release distance from where it started. If it was not calculated to pass directly through the lens center it was adjusted according to the applicable geometry. By then taking these point-angle release coordinates the images were projected from the image plane to the camera as would be done in a real, focused, PIV system.

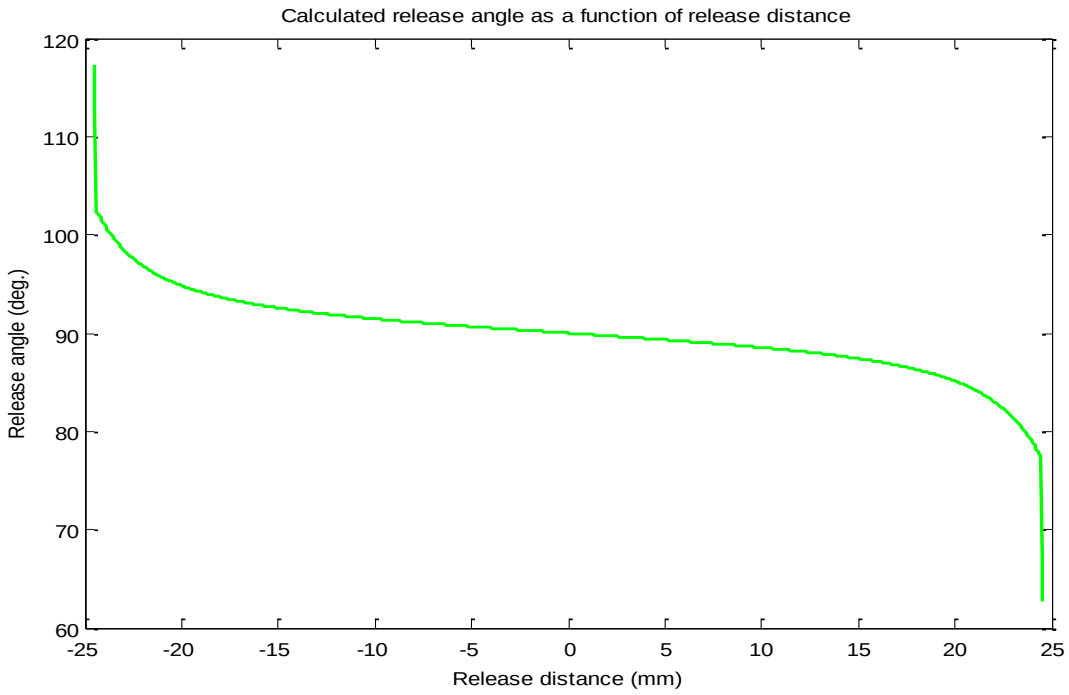

Figure 15: Release angle for focused image for the setup parameters given in Table 1.

4.2 Bubble Distortion Effects

The next portion of identifying the distortion effects caused by the bubbles was to create the images as seen through the distorted visualization system. It was rather apparent that when different bubbles were included in the field of view the image was transformed dramatically. The following, Figure 16 through Figure 18, show the distortion profiles of some representative small and large bubbles for

different bubble locations. For the purpose of calling out portions of lost data due to reflection, the portions that were calculated to be internally reflecting were assigned final angle values of 90 deg. for all angles and 90 mm for all distances. In general, it can be seen that for bubbles in the bulk fluid there were side portions at the edges of the bubble profile where the light would be internally reflected and lost. When inside the range of internal reflection it can be seen that the image is distorted rapidly as the location of intersection with the bubble is moved towards the edges of the bubble. When a bubble overlaps the viewing plane it can be seen, as shown in Figure 17, that the image lost additional portions at the middle section where the intersection limited so that no light be released due to the bubble phase being present in the visualization plane. With slug bubbles this effect is even more prominent since the slug bubble takes out a large portion of the potential release area.

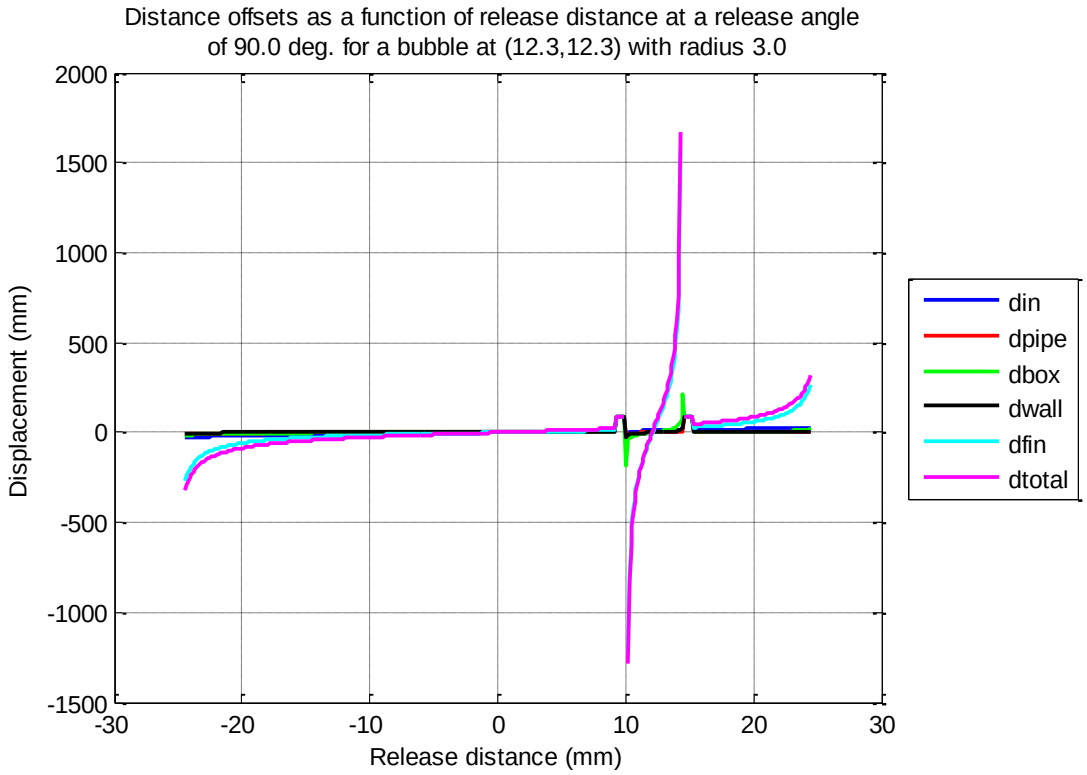

Figure 16: Distance offset as a function of release perpendicular to the viewing plane and a small bubble in the bulk fluid. Note the large slope and portions that have internal reflection.

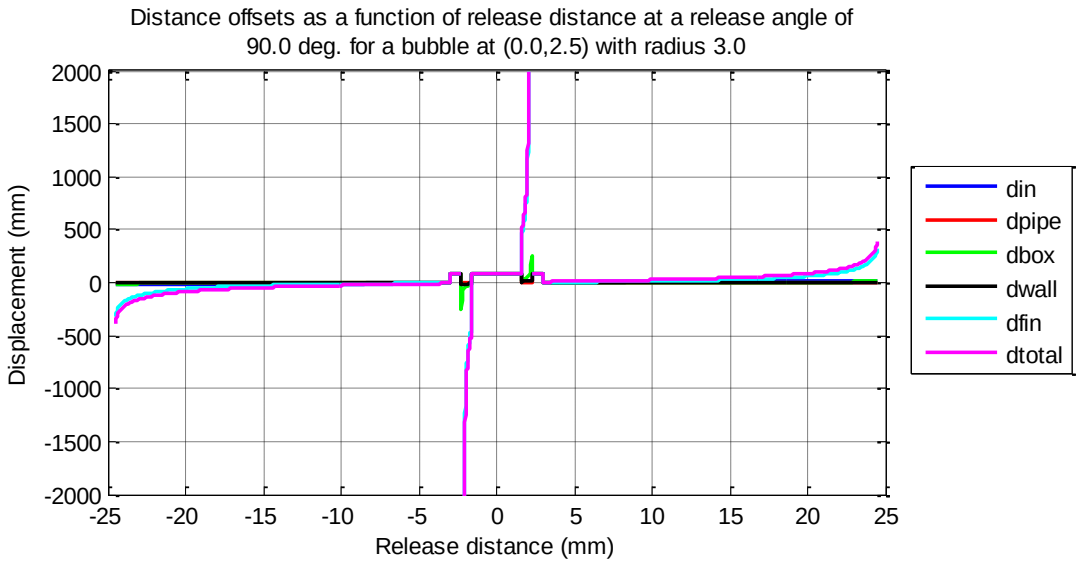

Figure 17: Distance offset as a function of release perpendicular to the viewing plane for a small bubble in the plane. Note the portion that is not emitting light in the center.

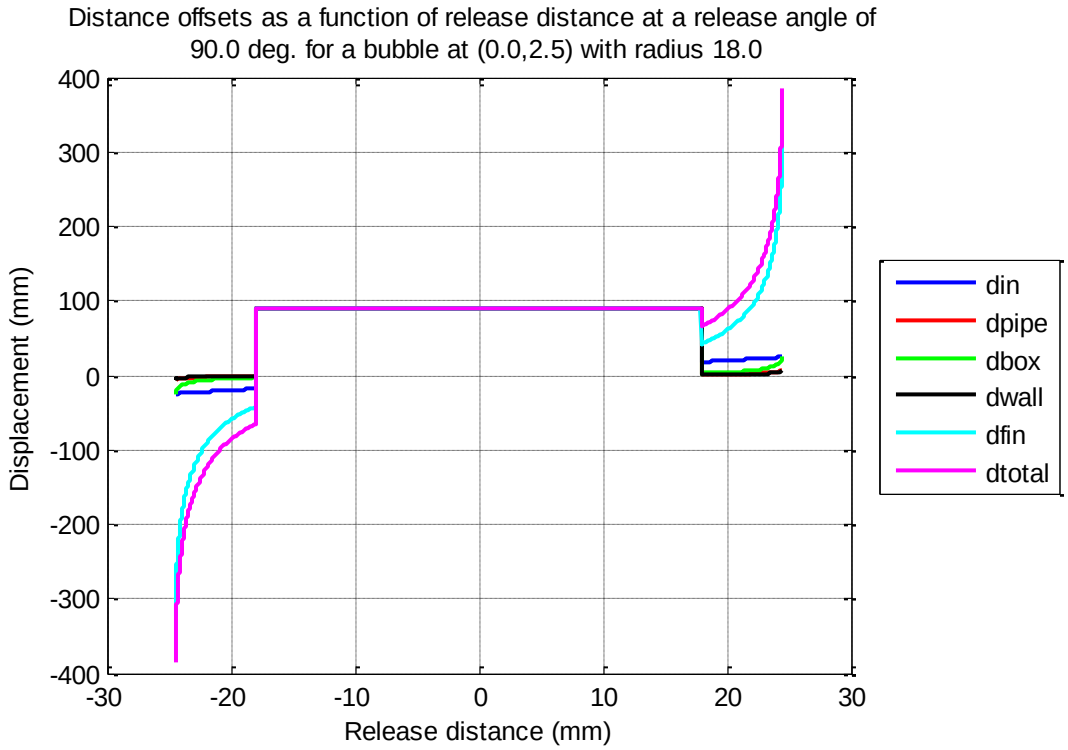

Figure 18: Distance offset as a function of release perpendicular to the viewing plane for a slug bubble nearly centered in the tube. Note the large portions that are not emitting light.

Figures 19 and 20 below show the steps used in the calibration of the distorted images. The process of calibration started by calculating the image that was transferred through the system from the

release profile described on page 12. The calculated image distances were then subtracted from the known image distances to create a calibration profile.

$$\text{Calibration} = \text{Release} - \text{Undistorted} \tag{38}$$

Next this calibration was applied to the distorted images using the following equation.

$$\text{CalibratedImage} = \text{Distorted} + \text{Calibration} \tag{39}$$

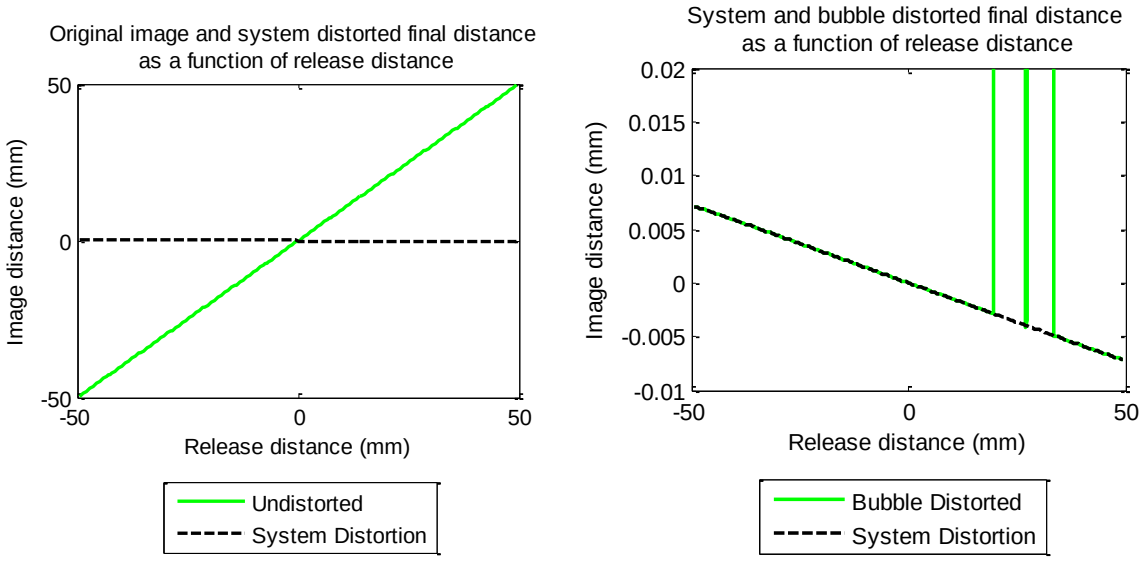

Figure 19: Calibration profile and distorted image calibration for large pipe.

As this error analysis was computed for the various bubble profiles it was found that the errors that were caused by the different geometries were very small compared to the rest of the data. By looking closely at the sites where the distortions are actually transmitted, in the middle of the bubble, it is possible to see that the error is smaller than the bubble is wide so there is no way that the focused image could distort out of bubble. The calibrated image of a representative 3mm bubble in the bulk fluid is shown in Figure 20. The error of interest is shown in Figure 21. Note the magnitudes of the axes. The trend that the calculated error was smaller than necessary to distort out of the bubble held for all instances tested here.

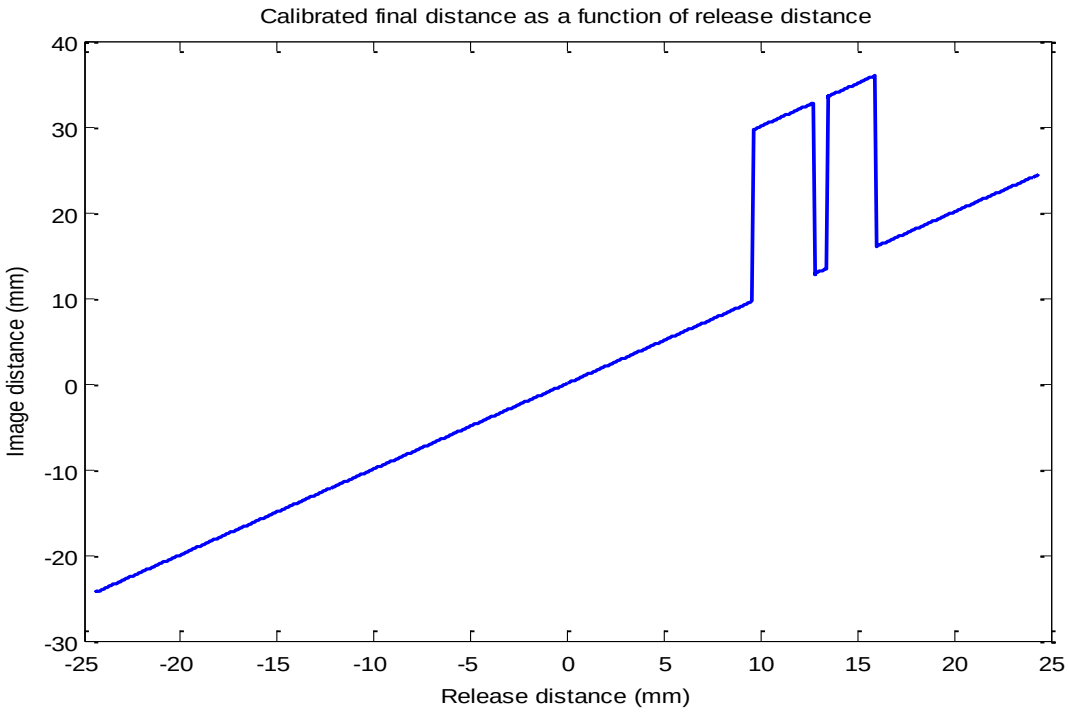

Figure 20: Calibrated image for a 3mm bubble at (12.5mm, 12.5mm).

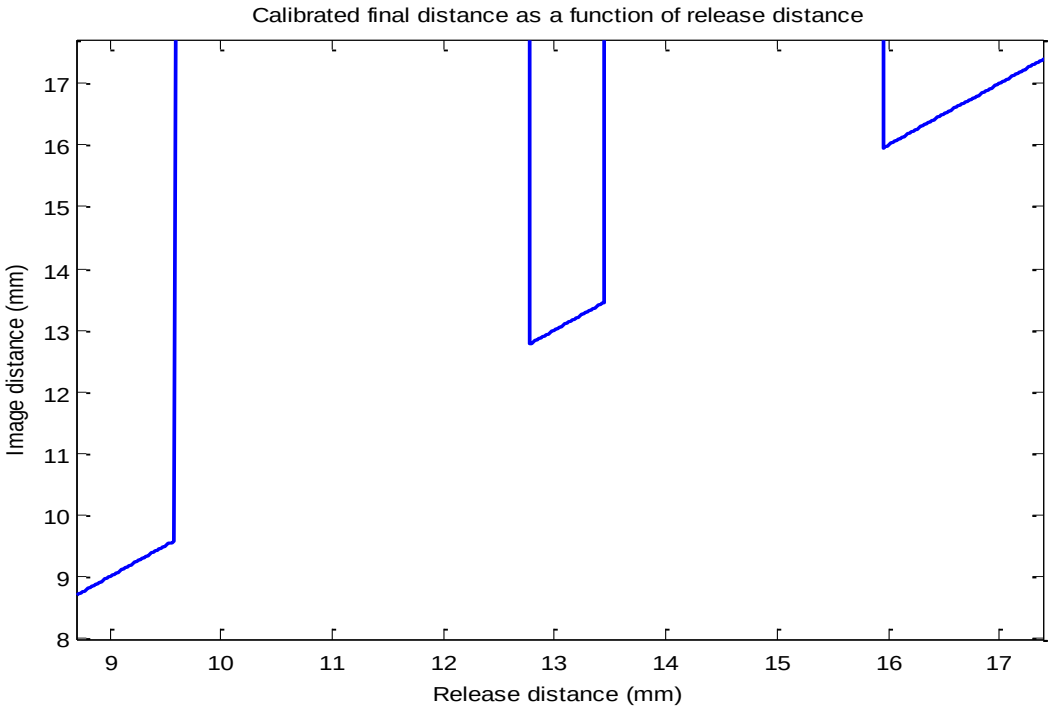

Figure 21: Close up of key portion of calibrated image for a 3mm bubble at (12.5mm, 12.5mm).

4.3 Error Approximation

From this analysis it can be seen that the error in the system due to inclusions of bubbles in the viewing window would not be significant. The points that would show error would be enclosed in an identifiable region. If calculation were required of particles that were knowingly obscured one could input the parameters of the inclusions they are interested in and run the program. By taking the maximum potential displacement extremes one could get the distance and subsequent velocity error in the measurements. Figure 22 shows a representative error plot that was created in the program using the equation

$$\mathrm{Imageerror} = \mathrm{CalibratedImage} - \mathrm{Release} \tag{40}$$

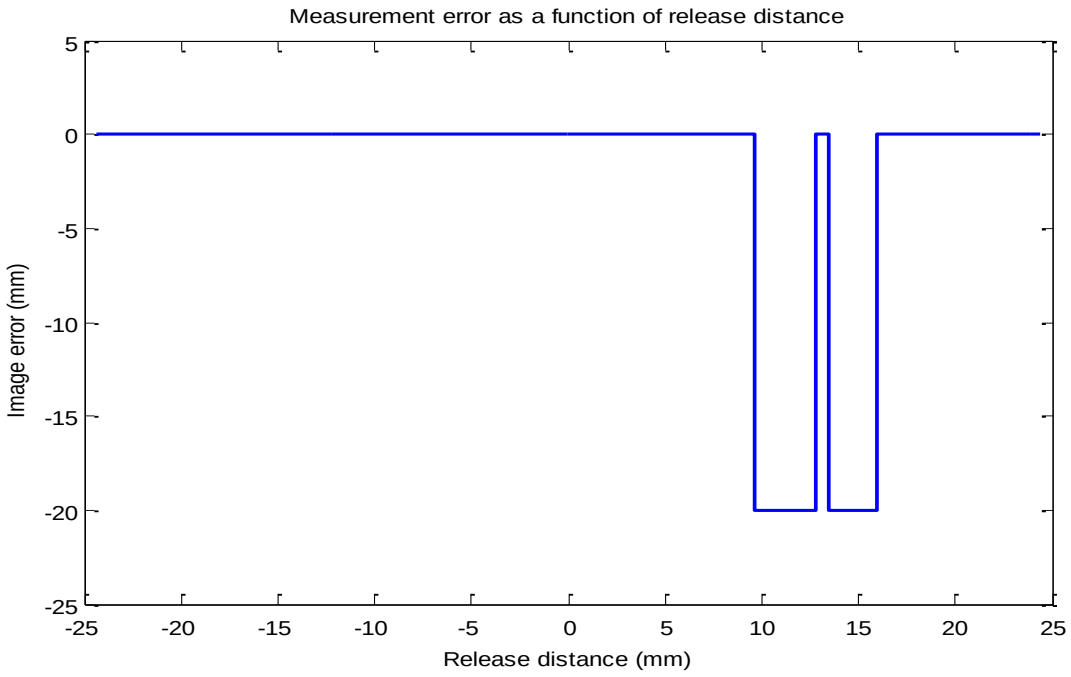

Figure 22: Measurement error as a function of release distance for a small bubble in the fluid. Note the obvious error points are artificially created in the code and actually represent lost data, not error. Bubble has 3mm radius.

To get a good sense of the amount of error this image needs to be scaled to magnify the true error. This is done in Figure 23 which shows that the error is less than the size of the bubble and thus will be easy to identify due to its location.

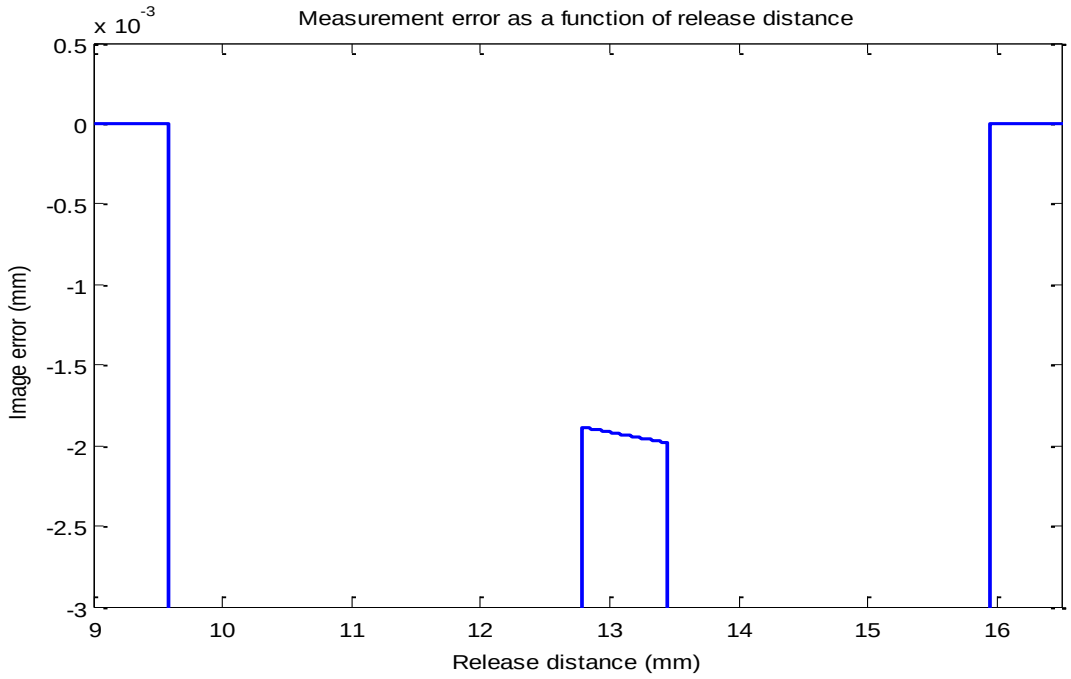

Figure 23: Close up of critical portion of measurement error as a function of release distance for a small bubble in the fluid. Bubble has 3mm radius.

Looking at other size bubbles provides similar results as can be seen in Figure 24 and Figure 26. Note that for the larger bubbles the amount of error is greater, however, since it is a larger bubble it still would be within the bounds of itself.

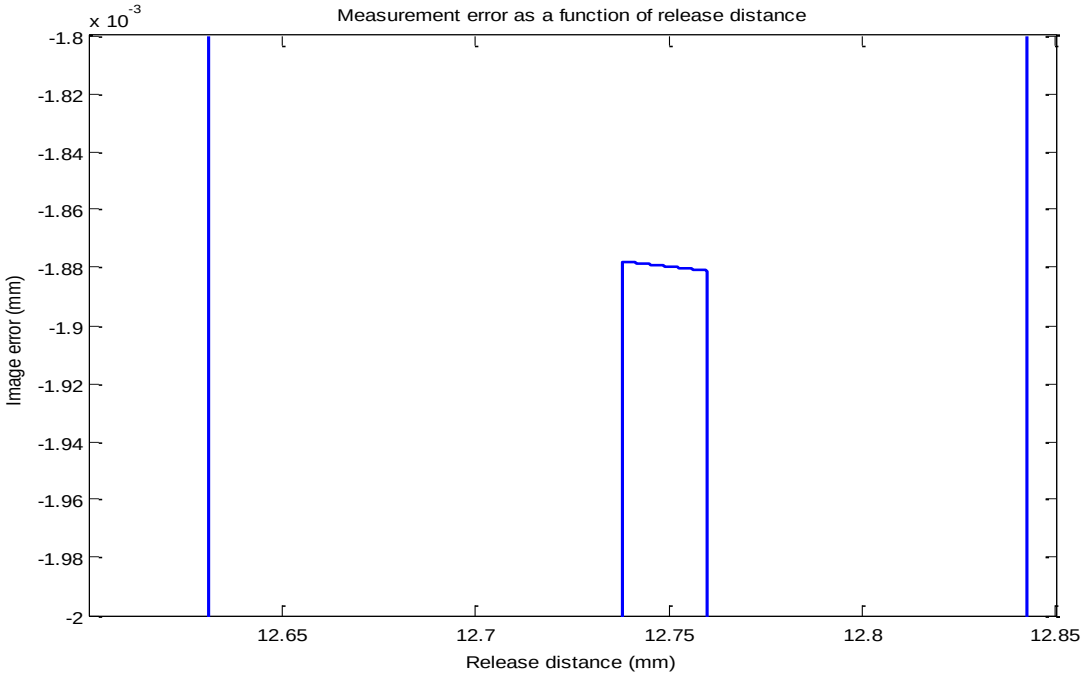

Figure 24: Measurement error distance as a function of release for a smaller bubble in the fluid.

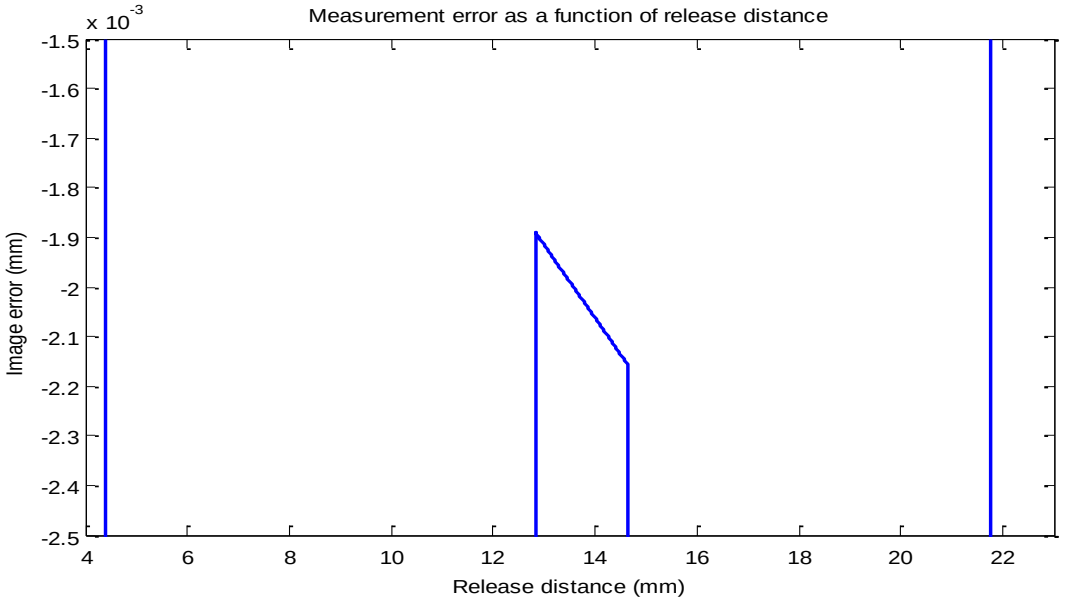

Figure 25: Measurement error distance as a function of release location for a larger bubble in the bulk fluid.

5. Questionable Results

From the analysis that has been done, it appears as though PIV users should not have much of an issue determining the locations of the particles due to measurement error induced by refraction in the bubbles. The window for which a particle must land in order to be transmitted through a distortion by a uniform circular bubble is quite small compared to the geometry of the system. From Figure 23 it can be seen that this window is right around 0.5mm for the bubble presented here, which is only 1/100 of the image window which is 49mm. If you compound these bubbles as is seen in real flows the probability of error would initially increase with the increase in void fraction, but at a certain point the addition of more bubbles would increase their probability of overlap, which in turn would most likely decrease the probability of light passing through to the camera. In addition, even if these particles were to be refracted and distort the image rather than simply obscure it, they would never have enough of a path difference to be sent out of the boundaries of the bubble itself. Thus, by identifying where the bubbles are would give you the entire area of possible image distortion.

In real application however, these results do not seem to account for all components of the distortion. Images, like the one shown below in Figure 26: Bubbly flow captured by Xinquan Zhou. This demonstrates the real distortions that are seen in two phase flow measurements. Note: the top image is calling out what appears to be a few particles that have refracted through an oblong bubble. The bottom right shows a bubble that looks to be distorted outward by a fairly round bubble and is showing the halo of no light. The bottom left shows the mystery light that should not be there according to this analysis., which was captured using the setup described previously by Xinquan Zhou, shows much more apparent distortion. The void areas that were predicted from this analysis are not obvious as was expected. Potentially this could be due to the more advanced refractions from multiple bubbles, but based on the results above it seems more likely that there is another component at work in the distortion of the images.

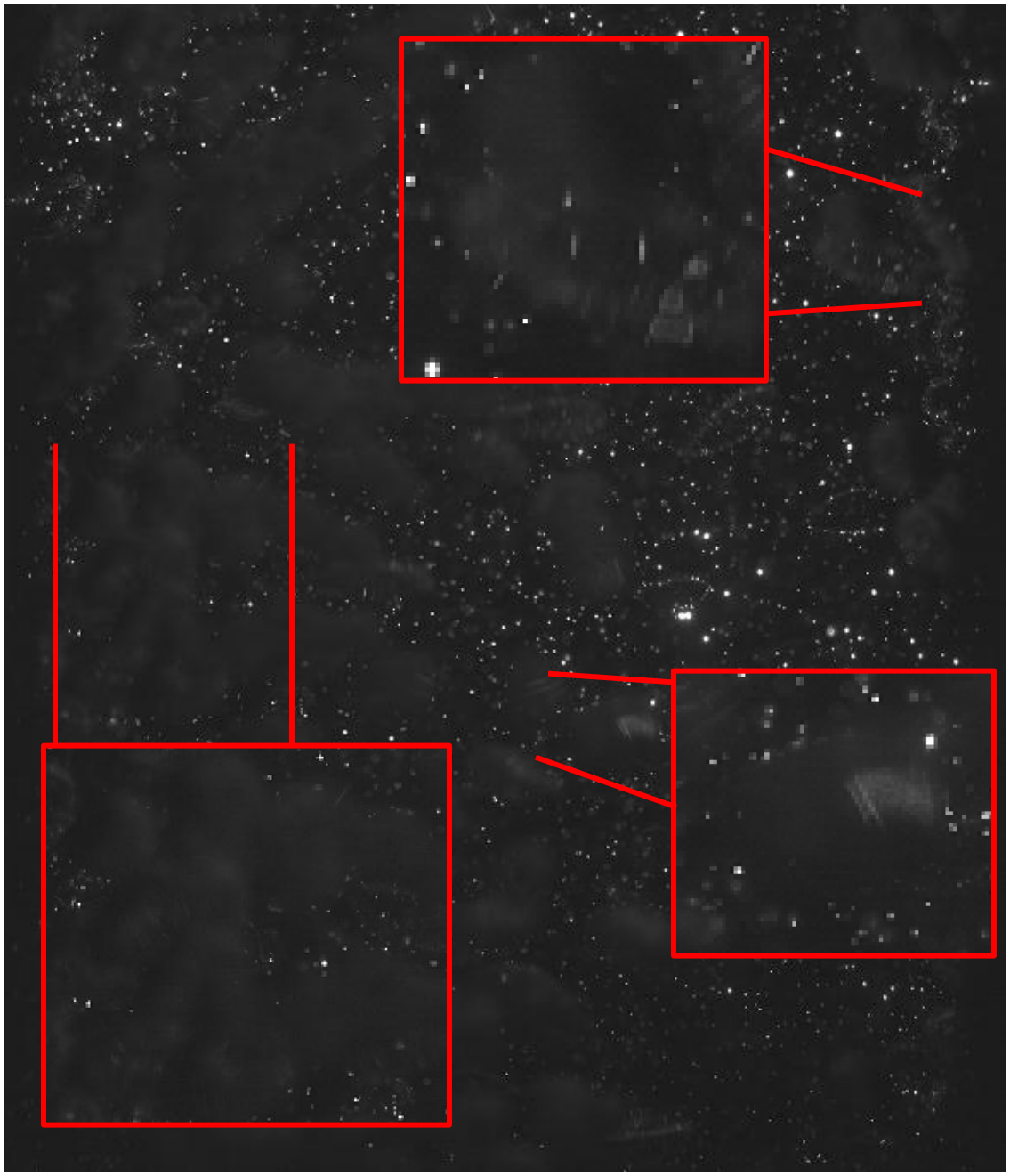

Figure 26: Bubbly flow captured by Xinquan Zhou. This demonstrates the real distortions that are seen in two phase flow measurements. Note: the top image is calling out what appears to be a few particles that have refracted through an oblong bubble. The bottom right shows a bubble that looks to be distorted outward by a fairly round bubble and is showing the halo of no light. The bottom left shows the mystery light that should not be there according to this analysis.

One possible reason for this unaccountable distortion image lies in the basics of refraction itself.

For this analysis it was assumed that any light that intersected a bubble or structure and was internally

reflected was lost to some unknown location. Also, it did not investigate the light reflection that always occurs coupled to light refraction. This could potentially account for the extra light which is seen. In theory, due to the transparency of the materials and the surface geometries that the light would be reflecting off of, the light would be naturally dispersed, potentially causing the glow that is seen. In theory, as with refraction, the images could be reflected to points that do not correspond to their true location, causing errors, however, in practice the probability of the image being concentrated while being reflected are much lower than if it is refracted. This would naturally reduce the intensity of the light as it is reflected around, plus, due to the nature of the reflected portion of the light, each reflection would reduce the reflected ray's intensity by a certain factor.

Putting all this together leads to the concept that the results from this project can in fact be correct despite not being able to account for the real images being distorted, as they are. The distortion component of more interest is probably the reflective one, as this component should be much easier to have occurred since internal reflection is less of a worry, and also should naturally disperse so as to be hard to measure. Both components do seem unlikely to cause phantom particle images which could be interpreted incorrectly.

6. Conclusion

The goal of this project was to investigate the distortion errors in PIV measurement of bubbly flow due to the bubble interactions. Tracing images through the system by geometrically calculating the refraction of light as it passed through the various constituent boundaries allowed for investigation of the refractive effects on the distortion of light. By first finding the image transfer through a system with no bubbles a calibration profile was created. Applying this calibration profile to images with bubble inclusions allowed for analysis of the error in the distorted images. This error was found to be smaller than the bubbles which caused the errors, and was centered on the previously mentioned bubbles in

such a way that the error from light refraction would be contained within the boundaries of the bubbles in the captured image. Since the error would be contained in the bubble it could be addressed.

In practice these results don't seem correct as there appears to be some manner of distortion in images taken of low and high void fractions. A potential explanation for this is that, in the formulation of this analysis the components of light that would be reflected or internally reflected were assumed to be lost. If these rays are included in the analysis they would probably have a diffractive nature and scatter rapidly. Also, the reflected portions would inherently be less intense since only a small portion would get reflected and most of the intensity would be transmitted through the bubble. Despite these trends to scatter and decrease in intensity it is possible if multiple conditions were met that the image could distort back to the camera and in fact create an error in the free field. Because of this potential for distortion from reflection a further analysis should be considered.

6.1 Contributions

This project has successfully created a program for estimating the error in flow measurements of two phase flows with interfering bubbles due to refraction. It has been found that the amount of measurement error is mainly a function of the assumed bubble radius, location of the bubble in the system and system geometry. It was found that in single camera PIV it would be reasonably identifiable where error could occur for all size bubbles as they would all have a characteristic marker of a bubble in the image which would identify the potential for a distortion error.

6.2 Additional Applications

By inverting the phase that is assumed to emit the light this analysis can be quickly modified to analyze the error in the gas region of the flow using PIV techniques with a seeded gas phase. Further, the nature of the analysis allows the programs which were generated to be applied to other geometries

that follow basic tube and viewing box geometry conventions. By modifying the material definitions and the thicknesses of the pipe and box walls this analysis could be further expanded to analyze unboxed pipe flow, inverted flows and free flows with optical distortion obstructions. For some of these cases the code would need to be modified to check for more cases of reflection that were not needed for this particular setup. If redone for these cases a step by step work through of all constituent equations would need to be done to make sure the sign convention errors and trigonometric anomalies were caught.

6.3 Future Work

This project has completed the analysis of the light travel through the system using numerical path tracking and geometrical relationships. In order to verify these results it would be ideal to take raw and calibrated PIV images of some known geometry and compare them to the results found in this investigation. The calibrated error method could be used for both instances. If the results of this experiment are accurate they should be able to recreate the experimental results with very little error.

In addition, as a second approach to the problem, this task could be approached using compound lens systems to directly recreate the geometry under study. This would provide a second method for verifying the results of this experiment. Depending on how this was done, redoing the analysis may provide a faster pathway for calculating the results. Finally, since the results cannot fully describe the distortion of the light in the PIV visualization the test should be expended to investigate the reflective distortion effects.

Bibliography

1003012_ImagingTools_D80.pdf. (2011, August 8). Gottengen, Germany.

Arroyo, M., & Greated, C. A. (1991). Stereoscopic particle image velocimetry. *Meas. Sci. Technol. 2 (1991)*, 1181-1186.

Cenedese, A., & Paglialunga, A. (1989). A new technique for the determination of the third velocity component with PIV. *Exp. Fluids 8*, 228-230.

Dinkelacker, F., Schafer, M., Ketterle, W., & Wolfrum, J. (1992). Determination of the third velocity component with PTA using an intensity graded light sheet. *Experiments in Fluids 13*, 357-359.

Gui, L., Lindken, R., & Merzkirch, W. (1997). Phase separated PIV measurements of the flow around systems of bubbles rising in water. Essen , Germany.

Hopkins, L. M., Kelly, J. T., Wexler, A. S., & Prasad, A. K. (n.d.).

Zhou, X., & Sun, X. (2011). Velocity Measurements in Gas-liquid Two-phase Slug and Churn-turbulent Flows. *2011 ANS Annual Meeting.* Hollywood, Florida.

Appendix A: Codes

Code 1: Full model for calculating final angles and distances based on given release distances and angles

for given inputs of bubble location and size, setup geometry and material properties.

```
%Joshua Jones
%Honors Research
%Full model of box enclosed pipe containing single phase water
%this program traces the light travel through the system from its
point of
%origin and inintial angle to give the distance at which it hits a
viewing
%plane at dc.
%Angles are in degrees
clear all
close all
clc

%%%%% Bubble Parameters
dby=2.5; %Bubble center from plane distance
dbx=0; %Bubble in plane center distance
rbub=18; %Radius of Bubble


%%%%% SETUP GEOMETRY
R=24.5;  %pipe radius
dwi=76; %distance from centerline to inside of wall
dc=600; %distance from wall to camera
fc=0.0127;  %distance past focal length
df=dc+fc;
tp=8.5; %thickness of pipe wall
tw=17; %wall thickness
dentire=dwi+dc+tw;


%%%%%% Indicies of Refraction
nw=1.33; %water
np=1.55; %plastic
na=1.00; %air
n1=nw;
n2=np;
n3=nw;
n4=np;
n5=na;
nbub=na;
chipdiam=0.0151552; %view size on chip
res=0.01; %release resolution

if rbub<=0 %There is no bubble
    fprintf ('There is no bubble\n')
```

```
else
    if rbub<=dby
    fprintf ('The bubble is in the bulk fluid\n')
    elseif rbub>=dby && dby>=0
    fprintf('The bubble is mostly in front of the viewing plane\n')
    elseif dby>=(-rbub) && dby<0
    fprintf('The bubble is mostly behind the viewing plane\n')
    elseif dby<=(-rbub)
    fprintf('The bubble is more than a bubble radius behind the
plane\n')
    elseif dby>=R %Bubble definitely intersects the pipe
    fprintf ('The bubble is not allowed to be in the pipe wall\n')
    end
end

%Index variable initiation
n=1;
neg=0;
IR=0;

%Varied conditions, suppress one and use as variable in the loop also,
set
%if x is defined here set angle=0
%P=0; %initial in plane distance from center of pipe [0,R]
angle=0;
x=90; %initial release angle [0,180]

%Loop Calculations
for P=-R+res:res:R-res
    IR=0;
    nopart=0;
    thetaio(n)=x;                  %initial release angle
    releaseio(n)=P;                 %initial release distance
    %Bubble shadow determination

    if rbub<=dby %The bubble is in the bulk fluid
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dright=dbx-abs(dby/tand(180-x))+abs(rbub*cosd(x-90));
        elseif x<=90
            dleft=dbx+abs(dby/tand(x))-abs(rbub*cosd(90-x));
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
        end
        if -R<=P && P<=dleft
            P=P;
            thetai(n)=thetaio(n);
        elseif dright<=P && P<=R
            P=P;
            thetai(n)=thetaio(n);
        elseif dleft<P && P<dright
            [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
        end
```

```
    elseif rbub>=dby && dby>=0 %The bubble is mostly in front of the
viewing plane
        %sagitta chord length calculation
        s=rbub-abs(dby);
        l=sqrt(2*rbub*s-s^2);
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<P && P<dright
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
                fprintf('right shadow1\n')
            elseif dleft<P && P<dileft
                fprintf('left shadow1\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
            end
        elseif x<=90
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif dileft>P && P>dleft
                fprintf('left shadow\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
            elseif diright<P && P<dright
                fprintf('right shadow\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
            end
        end
```

```
    elseif dby>=(-rbub) && dby<0 %The bubble is mostly behind the
viewing plane
        %sagitta chord length calculation
        s=rbub-abs(dby);
        l=sqrt(2*rbub*s-s^2);
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif dleft<P && P<dileft
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
            end
        elseif x<=90
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dileft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<P && P<dright
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,nw,na);
            end
        end

    elseif dby<=(-rbub) %Bubble is more than a bubble radius behind
the plane
        P=P;
        thetai(n)=x;

    elseif dby>=R %Bubble definitely intersects the pipe
    end


    %Check which side the point is released, forrect if needed
```

```
    if P<0
        neg=1;
        P=-P;
        thetai(n)=180-thetai(n);
    end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
    %Pipe Calculations
    alpha(n)=asind(P/R*sind(thetai(n))); %pipe wall intersection angle
    if IR==1
        alpha(n)=0;
    end
    beta(n)=asind(n1/n2*sind(alpha(n))); %pipe wall inner refracted
angle
    gamma(n)=asind(R/(R+tp)*sind(180-beta(n))); %pipe wall inner
intersection angle
    if abs(n2/n3*sind(gamma(n)))<=1
        delta(n)=asind(n2/n3*sind(gamma(n))); %pipe wall refracted
angle
    else
        delta(n)=90;
        fprintf('the light internally reflected at the pipe when
released from %.2f at an angle of %.2f\n',releaseio(n),thetaio(n))
        IR=1;
    end

    phi(n)=180-thetai(n)-alpha(n);
    if IR==1
        phi(n)=90;
    end
    din(n)=R*cosd(phi(n));
    zeta(n)=beta(n)-gamma(n); %angle offset from center of pipe due to
pipe thickness
    dpipe(n)=(R+tp)*cosd(phi(n)+zeta(n))-din(n);%Horizontal offset in
the pipe wall
    dpo(n)=(R+tp)*sind(phi(n)+zeta(n));  %Vertical distance to exit
point of pipe

    %Wall Calculations
    epsilon(n)=90-phi(n)-zeta(n)-delta(n); %box wall incident angle
    eta(n)=asind(n3/n4*sind(epsilon(n))); %box wall inner refracted
angle
    if abs(n4/n5*sind(eta(n)))<=1
```

```
        xi(n)=asind(n4/n5*sind(eta(n))); %box wall outer refracted
angle
    else
        xi(n)=90;
        fprintf('the light internally reflected at the wall when
released from %.2f at an angle of %.2f\n',releaseio(n),thetaio(n))
    end
    dbox(n)=(dwi-dpo(n))*tand(epsilon(n));    %Horizontal offset in
box
    dwall(n)=tw*tand(eta(n));                 %Horizontal offset in box
wall
    dfin(n)=dc*tand(xi(n));                %Horizontal offset outside
box
    dtotal(n)=din(n)+dpipe(n)+dbox(n)+dwall(n)+dfin(n); %Total
horizontal offset error

    %Correct the data if flipped
    if neg==1;
        P=-P;
        alpha(n)=-alpha(n);
        beta(n)=-beta(n);
        gamma(n)=-gamma(n);
        delta(n)=-delta(n);
        epsilon(n)=-epsilon(n);
        eta(n)=-eta(n);
        xi(n)=-xi(n);
        din(n)=-din(n);
        dpipe(n)=-dpipe(n);
        dpo(n)=-dpo(n);
        dbox(n)=-dbox(n);
        dwall(n)=-dwall(n);
        dfin(n)=-dfin(n);
        dtotal(n)=-dtotal(n);
        neg=0;
    end

    %Make data obvious if it intersected a bubble
    if IR==1
        alpha(n)=90;
        beta(n)=90;
        gamma(n)=90;
        delta(n)=90;
        epsilon(n)=90;
        eta(n)=90;
        xi(n)=90;
        din(n)=90;
        dpipe(n)=90;
        dpo(n)=90;
        dbox(n)=90;
        dwall(n)=90;
        dfin(n)=90;
        dtotal(n)=90;
```

```
    end

    %Make if obvious if the light was emitted where there is no fluid
    if nopart==1
        alpha(n)=90;
        beta(n)=90;
        gamma(n)=90;
        delta(n)=90;
        epsilon(n)=90;
        eta(n)=90;
        xi(n)=90;
        din(n)=90;
        dpipe(n)=90;
        dpo(n)=90;
        dbox(n)=90;
        dwall(n)=90;
        dfin(n)=90;
        dtotal(n)=90;
    end
    n=n+1;
end

if angle==1
%Angle investigation
figure
plot(thetaio, alpha, thetaio, beta,'r', thetaio, gamma,'g', thetaio,
delta,'k', thetaio, epsilon,'c', thetaio, xi,'m','LineWidth',2)
str = sprintf('Angle at pipe outside as a function of input angle at a
release distance of %f mm',P);
title(str);
xlabel ('Release angle (deg)')
ylabel ('Final angle (deg)')
legend
('alpha','beta','gamma','delta','epsilon','xi','Location','EastOutside
')
grid on
figure
plot(thetaio, din, thetaio, dpipe,'r', thetaio, dbox,'g', thetaio,
dwall,'k', thetaio, dfin,'c', thetaio, dtotal,'m','LineWidth',2)
str = sprintf('Distance offsets as a function of release angle at a
release distance of %f mm',P);
title(str);
xlabel ('Release angle (deg)')
ylabel ('Displacement (mm)')
legend
('din','dpipe','dbox','dwall','dfin','dtotal','Location','EastOutside'
)
grid on
elseif angle==0
%Distance investigation
figure
```

```
plot(releaseio, alpha, releaseio, beta,'r', releaseio, gamma,'g',
releaseio, delta,'k', releaseio, epsilon,'c', releaseio,
xi,'m','LineWidth',2)
str = sprintf('Angle at pipe outside as a function of release distance
at a release angle of %.1f deg. for a bubble at (%.1f,%.1f) with
radius %.1f',x,dbx,dby,rbub);
title(str);
xlabel ('Release distance (mm)')
ylabel ('Final angle (deg)')
legend
('alpha','beta','gamma','delta','epsilon','xi','Location','EastOutside
')
grid on
figure
plot(releaseio, din, releaseio, dpipe,'r', releaseio, dbox,'g',
releaseio, dwall,'k', releaseio, dfin,'c', releaseio,
dtotal,'m','LineWidth',2)
str = sprintf('Distance offsets as a function of release distance at a
release angle of %.1f deg. for a bubble at (%.1f,%.1f) with radius
%.1f',x,dbx,dby,rbub);
title(str);
xlabel ('Release distance (mm)')
ylabel ('Displacement (mm)')
legend
('din','dpipe','dbox','dwall','dfin','dtotal','Location','EastOutside'
)
grid on
end
```

Code 2: Bubble function for spherical bubbles at given location and size with input light angle and distance. Output is shifted location and angle.

```
function [Pnew, thetanew,IR]=bubble(thetai,P,dbx,dby,rbub,n1,n2)

%this function adjusts the angle and release distance for light that
intersects a
%bubble of radius rbub with fluids with refractive indicies n1 and n2
IR=0;
if thetai>=90
%Angle calculations
cd=sqrt(dby^2+(dbx-P)^2);
L=acosd((dbx-P)/cd);
M=180-thetai-L;
N=180-asind(cd/rbub*sind(M));
H=180-N;
    if abs(n1/n2*sind(H))<=1
        I=asind(n1/n2*sind(H));
    else
        I=90;
        fprintf('the light internally reflected in a bubble when
released from %.2f at an angle of %.2f\n',P,thetai)
        IR=1;
    end
J=I;
K=asind(n2/n1*sind(J));
O=180-J-I;
thetanew=thetai-(180-O-H-K);

%Distance Calculations
s1=sqrt(cd^2+rbub^2-2*rbub*cd*cosd(H-M));
if O==180
    s2=2*rbub;
elseif O==0
    s2=0;
else
    s2=sind(O)/sind(I)*rbub;
end
x1=cosd(M+L)*s1;
y1=sind(M+L)*s1;
x2=sind(270-N-M-L-I)*s2;
y2=cosd(270-N-M-L-I)*s2;
x=x1+x2;
y=y1+y2;
xa=y/tand(thetanew);
Pnew=P+x+xa;



%Initial angle is towards the center of the pipe
```

```
elseif thetai<=90
    %initial variable adjustment to treat like its from the left for
    %calculation purposes
    P=2*dbx-P;
    thetai=180-thetai;
%Same analysis as before
%Angle calculations
cd=sqrt(dby^2+(dbx-P)^2);
L=acosd((dbx-P)/cd);
M=180-thetai-L;
N=180-asind(cd/rbub*sind(M));
H=180-N;
    if abs(n1/n2*sind(H))<=1
        I=asind(n1/n2*sind(H));
    else
        I=90;
        fprintf('the light internally reflected in a bubble when
released from %.2f at an angle of %.2f\n',P,thetai)
        IR=1;
    end
J=I;
K=asind(n2/n1*sind(J));
O=180-J-I;
thetanew=thetai-(180-O-H-K);

%Distance Calculations
s1=sqrt(cd^2+rbub^2-2*rbub*cd*cosd(H-M));
if O==180
    s2=2*rbub;
elseif O==0
    s2=0;
else
    s2=sind(O)/sind(I)*rbub;
end
x1=cosd(M+L)*s1;
y1=sind(M+L)*s1;
x2=sind(270-N-M-L-I)*s2;
y2=cosd(270-N-M-L-I)*s2;
x=x1+x2;
y=y1+y2;
xa=y/tand(thetanew);
Pnew=P+x+xa;
end
```

Code 3: Cmab and cmabfocused function. These are modified versions of Code 1 in function form and are used to calculate image transfer through system. The only difference between the two functions is that in cmabfocused the final horizontal offset dfin is output while for cmab the variable P, which at the time of output is the modified release distance, is output.

```
function
[dtotal,xi,dfin]=cmabfocus(P,x,dbx,dby,rbub,R,n1,n2,n3,n4,n5,nbub,tw,t
p,dwi,dc)
%Index variable initiation
n=1;
neg=0;
    IR=0;
    nopart=0;
    thetaio(n)=x;                  %initial release angle
    releaseio(n)=P;                 %initial release distance
    %Bubble shadow determination

    if rbub<=dby %The bubble is in the bulk fluid
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dright=dbx-abs(dby/tand(180-x))+abs(rbub*cosd(x-90));
        elseif x<=90
            dleft=dbx+abs(dby/tand(x))-abs(rbub*cosd(90-x));
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
        end
        if -R<=P && P<=dleft
            P=P;
            thetai(n)=thetaio(n);
        elseif dright<=P && P<=R
            P=P;
            thetai(n)=thetaio(n);
        elseif dleft<P && P<dright
            [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
        end


    elseif rbub>=dby && dby>=0 %The bubble is mostly in front of the
viewing plane
        %sagitta chord length calculation
        s=rbub-abs(dby);
        l=sqrt(2*rbub*s-s^2);
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
```

```
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<P && P<dright
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
                fprintf('right shadow1\n')
            elseif dleft<P && P<dileft
                fprintf('left shadow1\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
            end
        elseif x<=90
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif dileft>P && P>dleft
                fprintf('left shadow\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
            elseif diright<P && P<dright
                fprintf('right shadow\n')
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
            end
        end

    elseif dby>=(-rbub) && dby<0 %The bubble is mostly behind the
viewing plane
        %sagitta chord length calculation
        s=rbub-abs(dby);
        l=sqrt(2*rbub*s-s^2);
        if x>90
            dleft=dbx-abs(dby/tand(180-x))-abs(rbub*cosd(x-90));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
```

```
            elseif -R<=P && P<=dleft
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif dleft<P && P<dileft
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
            end
        elseif x<=90
            dright=dbx+abs(dby/tand(x))+abs(rbub*cosd(90-x));
            dileft=dbx-l;
            diright=dbx+l;
            if dileft<=P && P<=diright
                P=0;
                thetai(n)=90;
                nopart=1;
            elseif -R<=P && P<=dileft
                P=P;
                thetai(n)=thetaio(n);
            elseif dright<=P && P<=R
                P=P;
                thetai(n)=thetaio(n);
            elseif diright<P && P<dright
                [P,thetai(n),IR]=bubble(x,P,dbx,dby,rbub,n1,nbub);
            end
        end

    elseif dby<=(-rbub) %Bubble is more than a bubble radius behind
the plane
        P=P;
        thetai(n)=x;

    elseif dby>=R %Bubble definitely intersects the pipe
    end


    %Check which side the point is released, forrect if needed
    if P<0
        neg=1;
        P=-P;
        thetai(n)=180-thetai(n);
    end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
    %Pipe Calculations
    alpha(n)=asind(P/R*sind(thetai(n))); %pipe wall intersection angle
    if IR==1
        alpha(n)=0;
    end
    beta(n)=asind(n1/n2*sind(alpha(n))); %pipe wall inner refracted
angle
    gamma(n)=asind(R/(R+tp)*sind(180-beta(n))); %pipe wall inner
intersection angle
    if abs(n2/n3*sind(gamma(n)))<=1
        delta(n)=asind(n2/n3*sind(gamma(n))); %pipe wall refracted
angle
    else
        delta(n)=90;
        fprintf('the light internally reflected at the pipe when
released from %.2f at an angle of %.2f\n',releaseio(n),thetaio(n))
        IR=1;
    end

    phi(n)=180-thetai(n)-alpha(n);
    if IR==1
        phi(n)=90;
    end
    din(n)=R*cosd(phi(n));
    zeta(n)=beta(n)-gamma(n); %angle offset from center of pipe due to
pipe thickness
    dpipe(n)=(R+tp)*cosd(phi(n)+zeta(n))-din(n);%Horizontal offset in
the pipe wall
    dpo(n)=(R+tp)*sind(phi(n)+zeta(n));  %Vertical distance to exit
point of pipe

    %Wall Calculations
    epsilon(n)=90-phi(n)-zeta(n)-delta(n); %box wall incident angle
    eta(n)=asind(n3/n4*sind(epsilon(n))); %box wall inner refracted
angle
    if abs(n4/n5*sind(eta(n)))<=1
        xi(n)=asind(n4/n5*sind(eta(n))); %box wall outer refracted
angle
    else
        xi(n)=90;
        fprintf('the light internally reflected at the wall when
released from %.2f at an angle of %.2f\n',releaseio(n),thetaio(n))
    end
    dbox(n)=(dwi-dpo(n))*tand(epsilon(n));    %Horizontal offset in
box
    dwall(n)=tw*tand(eta(n));                 %Horizontal offset in box
wall
    dfin(n)=dc*tand(xi(n));                %Horizontal offset outside
box
```

```
    dtotal(n)=din(n)+dpipe(n)+dbox(n)+dwall(n)+dfin(n); %Total
horizontal offset error

    %Correct the data if flipped
    if neg==1;
        P=-P;
        alpha(n)=-alpha(n);
        beta(n)=-beta(n);
        gamma(n)=-gamma(n);
        delta(n)=-delta(n);
        epsilon(n)=-epsilon(n);
        eta(n)=-eta(n);
        xi(n)=-xi(n);
        din(n)=-din(n);
        dpipe(n)=-dpipe(n);
        dpo(n)=-dpo(n);
        dbox(n)=-dbox(n);
        dwall(n)=-dwall(n);
        dfin(n)=-dfin(n);
        dtotal(n)=-dtotal(n);
        neg=0;
    end

    %Make data obvious if it intersected a bubble
    if IR==1
        alpha(n)=90;
        beta(n)=90;
        gamma(n)=90;
        delta(n)=90;
        epsilon(n)=90;
        eta(n)=90;
        xi(n)=90;
        din(n)=90;
        dpipe(n)=90;
        dpo(n)=90;
        dbox(n)=90;
        dwall(n)=90;
        dfin(n)=90;
        dtotal(n)=90;
    end

    %Make obvious if the light was emitted where there is no fluid
    if nopart==1
        alpha(n)=90;
        beta(n)=90;
        gamma(n)=90;
        delta(n)=90;
        epsilon(n)=90;
        eta(n)=90;
        xi(n)=90;
        din(n)=90;
        dpipe(n)=90;
```

```
        dpo(n)=90;
        dbox(n)=90;
        dwall(n)=90;
        dfin(n)=90;
        dtotal(n)=90;
    end
    n=n+1;
end
```

Code 4: Program to determine release conditions and then apply those through modified versions of previous codes 1 and 2 in order to produce final images of actual image. Also calculates errors in the measurements for analysis of PIV error.

```
%Joshua Jones
%Image projection program
clc
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% SETUP VARIABLES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

%%%%% Bubble Parameters
dby=12.25; %Bubble center from plane distance
dbx=12.25; %Bubble in plane center distance
rbub=8; %Radius of Bubble

%%%%% SETUP GEOMETRY
R=24.5;  %pipe radius
dwi=76.2; %distance from centerline to inside of wall
dc=609.6; %distance from wall to camera
% fc=0.0127;  %distance past focal length
fc=0.1;
df=dc+fc;
tp=8.5; %thickness of pipe wall
tw=17; %wall thickness
dentire=dwi+dc+tw;
lensdiam=62;    %Diameter of lens


%%%%%% Indicies of Refraction
nw=1.33; %water
np=1.55; %plastic
na=1.00; %air
n1=nw;
n2=np;
n3=nw;
n4=np;
n5=na;
nbub=na;
chipdiam=0.0151552; %view size on chip
res=0.01; %release resolution
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%% TELL BUBBLE LOCATION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
if rbub<=0 %There is no bubble
    fprintf ('There is no bubble\n')
else
    if rbub<=dby
        fprintf ('The bubble is in the bulk fluid\n')
    elseif rbub>=dby && dby>=0
        fprintf('The bubble is mostly in front of the viewing
plane\n')
    elseif dby>=(-rbub) && dby<0
        fprintf('The bubble is mostly behind the viewing plane\n')
    elseif dby<=(-rbub)
        fprintf('The bubble is more than a bubble radius behind the
plane\n')
    elseif dby>=R %Bubble definitely intersects the pipe
        fprintf ('The bubble is not allowed to be in the pipe wall\n')
    end
end


n=1;
%%Numerically construct the release angle array
%%%% IF GEOMETRY IS NOT SYMMETRIC THIS NEEDS TO BE CALCULATED ACROSS
THE
%%%%ENTIRE WINDOW!!! IF SYMMETRIC,ONLY NEED TO CALCULATE HALF
symmetric=1;
if symmetric==1
    left=0;
else
    left=-R+res;
end
for P=left:res:R-10*res;
    x=90;
    distance=cmab(P,x,dbx,dby,0,R,n1,n2,n3,n4,n5,nbub,tw,tp,dwi,dc);
    excape=1;
    while abs(distance)>=0.00000000001 && excape<5000;
        x=x-atand(distance/dentire);

distance=cmab(P,x,dbx,dby,0,R,n1,n2,n3,n4,n5,nbub,tw,tp,dwi,dc);
        excape=excape+1;
    end
    excape;
    if excape==5000
        fprintf ('the angle did not converge enough for release point
%.2f\n',P)
    end
    distance(n)=distance;
```

```
    if x>180
        theta(n)=x-180;
    elseif x<0
        theta(n)=x+180;
    else
        theta(n)=x;
    end
    Parray(n)=P;
    thetaorig(n)=x;
    n=n+1;
end
if symmetric==1
    thetaflip=180-flipdim(theta,2);
    theta=[thetaflip(1:(n-2)),theta];
end
figure
plot(Parray,thetaorig)
plot(Parray,distance);



n=1;
clear P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%% CALCULATE THE DISTORTED DISTANCES
%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%% CALCULATE ENTIRE PROFILE!!
%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
for P=-(R-10*res):res:(R-10*res)

[d(n),xi(n),Pnew]=cmab(P,theta(n),dbx,dby,rbub,R,n1,n2,n3,n4,n5,nbub,t
w,tp,dwi,dc);
    if abs(d(n))>=lensdiam;
        final(n)=20;
    elseif abs(d(n))>=0.00000000001;

[final(n),xi(n),dfin]=cmabfocus(P,theta(n),dbx,dby,rbub,R,n1,n2,n3,n4,
n5,nbub,tw,tp,dwi,dc);
        if dfin-d(n)==0
            final(n)=0;
        else
            x=90;

distance=cmab(Pnew,x,dbx,dby,0,R,n1,n2,n3,n4,n5,nbub,tw,tp,dwi,dc);
            excape=1;
            while abs(distance)>=0.00000000001 && excape<5000;
                x=x-atand(distance/dentire);
```

```
distance=cmab(Pnew,x,dbx,dby,0,R,n1,n2,n3,n4,n5,nbub,tw,tp,dwi,dc);
                excape=excape+1;
            end
            if excape==5000
                fprintf ('the angle did not converge enough for
release point %.2f\n',P)
            end
            if x>180
                thetanew=x-180;
            elseif x<0
                thetanew=x+180;
            else
                thetanew=x;
            end

[final(n),xi(n),dfin]=cmabfocus(Pnew,thetanew,dbx,dby,0,R,n1,n2,n3,n4,
n5,nbub,tw,tp,dwi,dc);
            fprintf('distorted point P=%.2f\n',P)
        end
    else

[final(n),xi(n)]=cmabfocus(P,theta(n),dbx,dby,rbub,R,n1,n2,n3,n4,n5,nb
ub,tw,tp,dwi,df);

    end
    if abs(final(n))>chipdiam
        final(n)=20;
    end

[original(n)]=cmab(P,theta(n),dbx,dby,0,R,n1,n2,n3,n4,n5,nbub,tw,tp,dw
i,df);
    release(n)=P;
    n=n+1;
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%%%%%%%%%%%%%%%%%%%%%% CALCULATE THE OVERLAY AND THE ERRORS
%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
figure
plot(release,theta,release,xi,'g','LineWidth',2)
title ('Calculated release angle as a function of release distance')
xlabel ('Release distance (mm)')
ylabel ('Release angle (deg.)')
legend ('input angle','final angle')
figure
plot(release,final,'g',release,original,'k--','LineWidth',2)
```

```
title ('System and bubble distorted final distance as a function of
release distance')
xlabel ('Release distance (mm)')
ylabel ('Image distance (mm)')
legend ('Bubble Distorted','System Distortion')

overlay=release-original;
figure
plot(release,(final+overlay),'LineWidth',2);
title ('Calibrated final distance as a function of release distance')
xlabel ('Release distance (mm)')
ylabel ('Image distance (mm)')

error=(release-(final+overlay))./release.*100;
figure
plot(release,error,'LineWidth',2);
title ('Measurement error as a function of release distance')
xlabel ('Release distance (mm)')
ylabel ('Image error (%)')

errordist=release-(final+overlay);
figure
plot(release,errordist,'LineWidth',2);
title ('Measurement error as a function of release distance')
xlabel ('Release distance (mm)')
ylabel ('Image error (mm)')
if errordist>10
    errordist=0;
end
maxerror=max(errordist);
minerror=min(errordist);
```